Calculation of two-dimensional pulse electron paramagnetic resonance (EPR) signals using Stochastic Liouville equation and application to a system of coupled electron spin ½ and nuclear spin ½ in a malonic acid crystal

Lin Li

A Thesis

In the Department

of

Physics

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Physics) at

Concordia University

Montreal, Quebec, Canada

June 2019

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Lin Li

Entitled: Calculation of two-dimensional pulsed electron spin resonanc (ESR) signals using Stochastic Liouville equation and applicaton to a system of coupled electron spin 1/2 and nuclear spin 1/2 in malonic acid crystal

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Physics)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____Chair
Dr. Yogendra Chaubey

____ External Examiner
Dr. Arthur Van der Est

____ External to Program
Dr. Gilles Peslherbe

____ Examiner
Dr. Mariana Frank

____ Examiner
Dr. Calvin Kalman

____Thesis Supervisor
Dr. Sushil K. Misra

Approved by ____
Dr. Valter Zazubovits, Graduate Program Director

August 6, 2019 ____
Dr. Andre Roy, Dean
Faculty of Arts and Science

ABSTRACT

Calculation of two-dimensional pulse electron paramagnetic resonance (EPR) signals using Stochastic Liouville equation and application to a system of coupled electron spin ½ and nuclear spin ½ in malonic acid crystal

Lin Li, Ph.D.

Concordia University, 2019

This dissertation is devoted to the subject of computations for pulse EPR by solving the Liouville von Neumann equation in Liouville space in the presence of relaxation, providing a comprehensive theoretical treatment, including a detailed description of relaxation processes.

The algorithm is illustrated here to simulate SECSY and echo-ELDOR signals for an electron-nuclear spin coupled system (electron spin S=1/2; nuclear spin I=1/2) in an irradiated malonic-acid crystal, taking into account spin-phonon modulation. The simulated results so obtained are compared with the experimental results of Lee *et al.* (1993). The experimental and simulated spectra are found to be in reasonably good agreement.

In addition, the developed algorithm is used to simulate the spectra for a specific case of the electron-electron dipolar interaction of two nitroxide radicals, each with electron spin S=1/2, nuclear spin I=1/2, for a proposed experiment, intended for distance measurements. The Pake doublets in the Fourier transform of both the signals calculated with and without relaxation correctly indicate the spacing of 10 MHz of dipolar coupling as assumed in the simulation.

This algorithm is also extended to calculate the effect of relaxation due to fluctuation of spin-Hamiltonian parameters by phonon modulation (Freed and Fraenkel, 1963; Slichter, 2013). From a comparison of the Figures showing the simulated SECSY and echo-ELDOR signals for an electron-nuclear spin-coupled system (S = I = ½) in an irradiated malonic acid crystal as calculated for $\tau_c = 10^{-5}, 10^{-6}, 10^{-7}$ and the fluctuation $\Delta(SHP) = 10\,\%, 1\%, 0.1\%$ of their static values as fitting parameters with the experimental results of Lee *et al.* (1993), it is seen that for the two cases considered (SECSY and echo-ELDOR) the best agreement is obtained for the following ranges: $\tau_c = 10^{-6}s$ - $10^{-7}s$ and $\Delta(SHP) = 0.1\ \%$ - $1.0\ \%$. The value of $\tau_c{\sim}10^{-7}s$ as calculated by Freed (1965) is found to be in the range of values found here that provide a good agreement of the calculated spectra with the experimental ones. In the light of these observations, it appears that the fluctuation model considered here, may indeed be refined further to reproduce the simulations to be in satisfactory agreement with the experiment.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Dr. Sushil K. Misra for his continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in my entire research work and writing of this thesis.

I would like to express my special gratitude to Dr. Mariana Frank and Dr. Calvin S. Kalman for providing me with encouragement during my studies at Concordia University.

I thank the Graduate Program Directors Dr. Laszlo Kalman, Dr. Pablo Bianucci and Dr. Valter Zazubovits, and Assistant to the Chair Madam Marie-Anne Cheong Youne for their help in my studies.

I also thank Mr. Hamid Reza Salahi for his careful checking of my calculations and constructive suggestions.

Financial assistance from Dr. Misra’s NSERC grant towards my bursary is gratefully acknowledged.

Last but not the least; I would like to thank my wife and children for their support throughout my study.

Table of Contents

List of Figures

List of Abbreviations

Echo-ELDOR - Echo Electron-electron Double Resonance

SECSY - Spin Echo Correlation Spectroscopy

DEER - Double Electron-Electron Resonance

EMR - Electron Magnetic Resonance

ENDOR - Electron Nuclear Double Resonance

EPR - Electron Paramagnetic Resonance

ESEEM - Electron Spin Echo Envelope Modulation

ESR - Electron Spin Resonance

FID - Free Induction Decay

FT - Fourier Transform

HF – Hyperfine

LVN - Liouville von Neumann

NMR - Nuclear Magnetic Resonance

PELDOR - Pulse Electron Double Resonance

SHP - Spin-Hamiltonian Parameters

SLE - Stochastic- Liouville equation

1. Introduction

Pulse electron paramagnetic resonance (EPR) is a powerful technique that can be exploited to reveal unique detailed insights into the electronic and geometric structures of environments of paramagnetic centers in biological, organic and inorganic systems. It can distinguish weak interactions between electron spins, as well as those between electron and nuclear spins, not resolved by continuous wave (CW) EPR. The wealth of information encoded in a pulse EPR spectrum of a paramagnetic system can be used to characterize the dynamics of spin probes embedded in diamagnetic systems. This requires suitable theoretical tools for interpreting the effects of molecular motions on the spectrum. Pulse EPR has a wide range of applications in biology, chemistry, physics and materials science, as described, e.g. by Schweiger and Jeschke (2001), Deligiannakis *et al,* (2000), Prisner *et al*, (2001), and Van Doorslaer *et al,* (2007) .

While excellent numerical simulation techniques for CW EPR have been developed and applied to experiments, see e.g. Hanson *et al,* (2004), Stoll *et al,* (2006), Misra (2011), the currently available simulation methods for pulse EPR have either limitations, or they are too general. They work well in some cases, but are usually limited to specific experimental cases, such as considering a single nucleus, an axial hyperfine-structure tensor, negligible nuclear quadrupole coupling tensor, an isotropic g-matrix, or an infinitely short pulse, see e.g. Vinck and Doorslaer (2004) García-Rubio *et al*, (2007). On the other hand, there are available two open-source packages, which are very general (Hogben *et al*, 2011; Pribitzer *et al*,2016) in that they do not deal with specific types of relaxation of spin-coupled systems in solids. To this particular purpose, it is necessary to exploit the Liouville von Neumann (LVN) equation numerically, as carried out in this thesis.

Hilbert-space formulation cannot be directly used to calculate relaxation effects, because the matrix elements of the relaxation superoperator require a pair of double indices, based on the eigenbasis of the Hamiltonian operator, which are not amenable to use in Hilbert space formulations, see e.g. Stoll (2009). Misra *et al,* (2009). Misra (2011, 2011a), Schwartz *et al,* (1982), Gamliel and Levanon (1995), Håkansson *et al*, (2013), Franck *et al,* (2015). Accordingly, a numerical simulation technique for calculating spin-correlation spectroscopy (SECSY) and echo electron-electron double resonance (echo-ELDOR) signals in pulse EPR experiments has been developed, here by treating the LVN equation as a matrix differential equation in Liouville space in order to take into account the relaxation effects rigorously. The LVN equation is an exact quantum-mechanical equation of motion for the density matrix. This equation is valid even for relatively fast random processes, and is therefore especially suitable for EPR, where the natural time scale is so short that the random processes are not usually fast on this time scale. It is, in fact, the time-dependent Stochastic Liouville equation, which includes, in addition to a relaxation term, a time-dependent but not stochastically time-dependent Hamiltonian, e.g. a Hamiltonian representing the pulses. Misra and Li (2018) carried out a solution of this problem, using the spin Hamiltonian specified in Lee *et al.* (1993). (This publication is included here as Appendix A).

A general theoretical approach is employed in this thesis to exploit the LVN equation numerically, which treats the evolution of the density matrix both in the presence and in the absence of a pulse, wherein the relaxation matrix can be numerically treated for the cases of (i) relaxation due to the spin-phonon modulation as carried out by Lee *et al*. (1993) to test the algorithm developed here, and (ii) relaxation due to the fluctuation of spin-Hamiltonian parameters by spin-phonon modulation. As for (ii), a framework for taking into account the fluctuation of spin-Hamiltonian parameters was outlined by Lee *et al.* (1993) for an electron-nuclear spin-coupled system with electron spin S=1/2 and nuclear spin I=1/2; however, they did not carry out any quantitative numerical calculations, which required evaluation of appropriate auto-correlation functions. A solution based on this approach is included in this thesis.

The organization of this thesis is as follows. The basic theory of pulse EPR is given in Chapter 2. The procedure for matrix solution of the LVN equation for pulse EPR experiments is developed in Chapter 3. Thereafter, the details of the calculation of spin-echo-correlation spectroscopy (SECSY) and echo electron-electron double-resonance (echo-ELDOR) signals, including selection of coherent electron pathways, are given in Chapter 4. The simulations of pulse EPR spectra for SECSY and Echo-ELDOR experiments for the cases discussed by Lee *et al* (1993) in a malonic acid single crystal for an electron-nuclear spin-coupled system with electron spin S=1/2 and nuclear spin I=1/2, and the calculation of echo-ELDOR signal for dipolar-coupled nitroxide radicals in malonic single crystal, which can be exploited to estimate distances between two nitroxide radicals, are described in Chapter 5. The relaxation due to the fluctuation of spin-Hamiltonian parameters by spin-phonon modulation is treated in Chapter 6. The concluding remarks and future perspectives are included in Chapter 7. The Matlab source code is included here in Appendix G.

2. Pulse electron paramagnetic resonance (EPR)

This chapter introduces a survey of relevant literature, involving the main concepts of pulse EPR to provide a background for the research work presented in this thesis dealing with the simulations of *spin-echo correlation spectroscopy (SECSY)* and *echo-electron double resonance (echo-ELDOR)* signals. (The important terms in the context of pulse EPR as described here are shown in italics to draw attention.) For a detailed description refer to e.g. Schweiger and Jeschke (2001), Atta-Ur-Rahman *et al.* (2015), Misra and Li (2018), PulsedEPR_wikipedia (2019).

2.1 Overview

EPR is a technique that can directly detect and quantify unpaired or odd electrons in atoms or molecules. Materials containing unpaired electrons are called paramagnetic materials. They exhibit net magnetic moments in an external magnetic field. EPR is also known as electron spin resonance (ESR). Although both names, EPR and ESR, are used in practice. A third name, electron magnetic resonance (EMR), has also been introduced, since the magnetic resonance is caused by the electron magnetic moment. Throughout this thesis, the term EPR is used.

EPR is a spectroscopic technique, which is widely used in biology, chemistry, medicine and physics to study atom or molecule systems with one or more unpaired electrons, see. e.g. Prisner *et al*. (2001), Misra *et al*. (2015), Åman *et al*. (2007). EPR and ENDOR provide information on the structure, dynamics and spatial distribution of paramagnetic materials due to the special relationship between magnetic parameters, electron wave functions and spin configurations of surrounding nuclei (Van Doorslaer *et al*., 2007).

Traditional continuous wave (CW) EPR methods, wherein the applied microwave frequency is kept fixed and the external magnetic field is swept, are limited in both spectral and temporal resolution. In Pulse EPR, the spectrum is obtained by applying a variety of pulse sequences to produce a two-dimensional time-domain spectrum, e.g. *spin-echo correlation spectroscopy (SECSY) and echo-electron double resonance (echo-ELDOR)*, using pulses of the same frequency, as simulated in detail in this thesis. (N.B. The use of "double" in echo-ELDOR does not imply use of two frequencies here.)

Blume (1958) first reported electron spin echo. It was produced in sodium in ammonia solution at its boiling point -33.8°C using the magnetic field of 62 mT and the frequency 17.4 MHz. In the same year, Gordon and Bowers (1958) reported the detection of microwave electron spin echoes, using frequencies of 23 GHz to excite dopants in silicon. As early as 1960s, the Mims (1965) team of Bell Laboratories conducted a large number of pulse EPR studies. However, in the first ten years, only a few groups were able to do research in this field, because of the expensive instruments, the lack of microwave components and the slow speed of digital electronic devices. In 1961, Mims, Nassau and McGee (1961) were the firsts to observe electron spin echo envelope modulation (ESEEM). Mims (1965) invented pulse electron nuclear double resonance (ENDOR) in 1965. In this experiment, pulse NMR transitions were detected by pulse EPR. Pulse ESEEM and pulse ENDOR are still important for studying the coupling between nuclear and electron spins.

2.2 Pulse EPR

Pulse EPR is an electron resonance technique, which investigates the alignment of the net magnetization vector of electron spins rotating in a constant external magnetic field, which is disturbed by short oscillating fields (usually microwave pulses). The resulting signal produced by the sample magnetization can be measured in time domain. The Fourier transform of this signal generates its EPR spectrum in the frequency domain. By the use of diverse pulse sequences, it is possible to have a detailed understanding of the structure and kinetic properties of paramagnetic compounds. Pulse EPR techniques, such as *electron spin echo envelope modulation (ESEEM)* or *electron nuclear double resonance (ENDOR)*, can reveal the interaction between electron spin and its surrounding nuclear spin, as well as with the environment.

The basic principle of pulse EPR is similar to that of pulse nuclear magnetic resonance (NMR). However, the relative magnitude and relaxation times are remarkably different in EPR from those in NMR. In EPR, the relaxation rate is faster than that in NMR. Since the magnetization is measured as a bulk property, its description by classical theory gives a more intuitive view. In order to better understand the concept of pulse EPR let us consider the behavior of the magnetization vector both in the laboratory and rotating frames. In the laboratory frame, for the present discussion, the static magnetic field, B_0, is assumed parallel to the z-axis and the microwave field B_1 parallel to the x-axis. When an electron spin is placed in a magnetic field, it experiences a torque that makes its magnetic moment precess around the magnetic field. The precession frequency is called Larmor frequency (Schweiger and Jeschke, 2001):

$$\omega_L = -\gamma B_0 \tag{2.1}$$

In Eq. (2.1), γ is the gyromagnetic ratio of the electron. The electron spin has two quantum mechanical states, one parallel and the other antiparallel to the direction of the magnetic field B_0. The Boltzmann distribution dictates the population to be more in the electron-spin state parallel to the magnetic field as it is lower in energy. This leads to a net non-zero magnetization.

The effect of the microwave field B_1 can be better understood in the *rotating frame*. In an EPR experiment usually a microwave resonator is used to generate a linearly polarized microwave field B_1, equivalent to two circularly polarized microwave fields rotating in opposite directions, perpendicular to the much stronger static magnetic field B_0. The rotating frame is fixed on one of these rotating B_1 components. First, it is assumed that the B_1 field is in resonance with the precession of the magnetization vector M_0, so that the Larmor frequency is equal to the frequency of the B_1 field:

$$\omega_L = \omega_0 \tag{2.2}$$

Therefore, in the rotating frame, the B_1 component and the precessing magnetization component appear to be stationary, leading effectively to the disappearance of B_0. Then, one only needs to consider B_1 and M_0. The M_0 vector is affected by the fixed microwave field B_1, which leads to another precession of M_0. This time, it moves around B_1 with the frequency ω_1:

$$\omega_1 = -\gamma B_1 \tag{2.3}$$

This angular frequency ω_1 is also known as *Rabi frequency*. Assuming that B_1 is parallel to the x-axis, the magnetization vector will rotate around the + x-axis in the zy-plane as long as the microwave field is on. The angle of rotation M_0 is called the tip angle theta, expressed as:

$$\theta = -\gamma |B_1| t_p \tag{2.4}$$

Here t_p is the duration of the B_1 field, which is also known as the pulse length. The pulses are marked by the rotation angle of M_0 that they cause. and by their direction when causing the rotation. For example, a +y π/2 pulse means that a B_1 pulse in the +y direction rotates M_0, originally in the +z direction, by the tip angle of π/2 to be along the -x axis. This implies that the final direction of the magnetization vector M_0 depends on the length, size and direction of the microwave pulse B_1. After this rotation of M_0, the sample emits microwaves, which are better understood in the laboratory frame of reference. For the case of resonance in a rotating frame, the magnetization appears to be stationary along the x- or y- axis after the pulse. In the laboratory frame, it rotates at the Larmor frequency in the xy plane. This rotation produces a signal that is maximum when the magnetization vector is exactly in the xy-plane. The microwave signal generated by this rotating magnetization vector is called *free induction decay (FID)* (Schweiger and Jeschke, 2001).

Non-resonant effects due to the presence of non-resonant frequencies present in the EPR spectrum lead to three main consequences. (i) The first one is better understood in the rotating frame. Now, a π/2 pulse leaves magnetization in the xy plane. But, since the frequency of the microwave field (and thus the rotating frame) is different from that of the precessing magnetization vector, the latter rotates in the xy-plane faster or slower than the microwave magnetic field B_1. The rotation rate is governed by the frequency difference Δω:

$$\Delta\omega = \omega - \omega_0 \tag{2.5}$$

When Δω = 0, the rotation speed of the microwave field is equal to that of the magnetization vector, then both of them seem to be stationary. If Δω > 0, the magnetization will rotate faster in the anticlockwise direction than the microwave component, and if Δω < 0, the magnetization will slow down, rotating in the clockwise fashion. This means that the individual frequency components of the EPR spectrum will appear as magnetization components rotating in the xy-plane with the frequency Δω. (ii) The second consequence appears in the laboratory frame, in which B_1 tips the magnetization differently out of the z-axis, because B_0 does not disappear completely when not at resonance due to the precession of the magnetization vector at Δω. This implies that the magnetization is now tipped by an effective magnetic field B_{eff}, which is the vector sum of B_1 and B_0, causing the magnetization to tip around B_{eff} at a faster effective rate ω_{eff}.

$$\omega_{eff} = (\omega_1^2 + \Delta\omega^2)^{1/2} \tag{2.6}$$

(iii) This leads directly to the third consequence that the magnetization cannot be efficiently tipped into the xy-plane, because unlike B_1, B_{eff} is not located in the xy-plane. The motion of the magnetization vector now defines a cone. That is to say that as Δω becomes larger, the magnetization is tipped less effectively into the xy-plane, and the FID signal decreases. In broad EPR spectra where Δω > ω_1, it is not possible to tip all the magnetization into the xy-plane to generate a strong FID signal. This is why it is important to maximize ω_1, which is equivalent to minimizing the length of the π/2 pulse, for broad EPR signals.

The magnetization, tipped into the xy-plane by a π/2 pulse, will decay and eventually return to the initial orientation and become aligned with the z-axis due to the interaction between the electron spin and its surroundings. The characteristic time for the magnetization to return to the z-axis by the relaxation process is described by the spin-lattice relaxation time T_1, as well as by the spin-spin relaxation time T_2, which is the time for the magnetization to vanish in the xy-plane. Spin-lattice relaxation is the manner by which the system returns to thermal equilibrium after being disturbed by the B_1 pulse, due to the interaction with the surrounding environment. The *spin-spin relaxation time*, also named *transverse relaxation time*, is related to the uniform and inhomogeneous broadening of the EPR signal. The inhomogeneous broadening is due to different spins undergoing inhomogeneous local magnetic fields due to different environments, resulting in a large number of spin packets, characterized by $\Delta\omega$ distributions. Due to the motion of the net magnetization vector, some spin packets decelerate due to the lower effective magnetic fields, while others accelerate due to the higher effective magnetic fields. This results in the fanning out of the magnetization vector, leading to the decay of the EPR signal. The decay of the transverse magnetization is caused by the homogeneous broadening of the spin packets, so that all spins in one spin packet will experience the same magnetic field and interact with each other. This results in mutual and random spin flips. The resulting fluctuations contribute to a faster fanning out of the magnetization vector.

The motion of the transverse magnetization is encoded with all the information about the frequency spectrum, which can be reconstructed using the time behavior of the transverse magnetization made up of y- and x-axis components, conveniently treated as the real (dispersion) and imaginary (absorption) parts of a complex quantity and use the Fourier theory to transform the measured time domain signal into the frequency-domain representation. Detection of both the absorption (imaginary) and dispersion (real) signals makes such an analysis possible.

Hahn echo. The FID signal decays away with time. For very wide EPR spectrum, this decay is rather fast due to the inhomogeneous broadening. One can use another microwave pulse to recover the lost signal to produce a *Hahn echo* (Hahn, 1950). This is accomplished as follows. After applying a $\pi/2$ pulse (90 degrees), the magnetization vector is transferred to the xy-plane and generates a FID signal. For inhomogeneous broadening different frequencies in the EPR spectrum cause the signal to "fan out", implying that the slow-spin data packets lag behind the fast-spin data packets. After a certain time t, a π pulse (180 degrees) is applied to the system to reverse the magnetization. This now causes the reverse, the fast-spin packets lag behind the slow-spin packets. When the signal is completely refocused at the time 2t, the exact echo caused by the second microwave pulse eliminates all the inhomogeneous-broadening effects. After all the spin packets bunched up into an echo, they dephase again just like a FID. This is equivalent to the spin echo exhibiting a reversal of FID followed by a normal FID. This can be Fourier-transformed to obtain the EPR spectrum. The longer the time t between the $\pi/2$ and π pulses, the smaller is the echo amplitude due to the effect of spin relaxation. This relaxation leads to an exponential attenuation of the echo height for different t times, whose attenuation constant is called the *phase memory time* T_M, which can have many contributions, such as transverse relaxation, spectral, spin and instantaneous diffusion. T_M can be measured directly by changing the time t between pulses.

2.3 Coherence pathways

It is helpful to think of pulse experiments in terms of coherence-transfer pathway diagrams. An electronic spin transition is labeled by the ‘p’ index, which can have the values −1, 0, or +1. In order to perform SECSY and ELDOR experiments, different coherences are combined in various ways, as described in the following chapters. They always start at p = 0 and end up at p = -1. The simulations in this thesis take into account these numerically by putting the relevant elements of the resulting density matrix equal to zero

Coherence may be considered as a generalized description of transverse magnetization; it corresponds to a transition between two energy levels. The magnitude of coherence in each spin system is governed by the coherence level. The difference in magnetic quantum number m_z of the two energy levels connected by the same coherence represents the *coherence level p*. The path describing the progress of a coherence order in a pulse sequence is called a *coherence transfer pathway*. It is possible to change the coherence level by applying a pulse. On the other hand, during the time interval between the applications of the pulses the coherence level does not change.

Transverse magnetization represents a particular type of coherence involving a change in the magnetic quantum numbers by ±1, representing the coherence level p = ±1. Each coherence level p is equal to the difference in the magnetic quantum numbers of the levels *r* and s in an EPR transition, i.e., the coherence level is p = $M_r - M_s$. Pulses cause transitions to occur between different coherence levels p. At thermal equilibrium, the change in *p* is zero, and it is at this point that the coherence transfer pathway is initiated by the application of a pulse. For signals to be detected by the receiver, the coherence transfer must end at a single quantum coherence, usually at $p = -1$.

2.4 Applications of pulse EPR and simulation software libraries

Electron Spin Echo Envelope Modulation (ESEEM) (Mims et al., 1961; Schweiger and Jeschke, 2001) and pulse Electron Nuclear Double Resonance (ENDOR) (Mims, 1965; Schweiger and Jeschke, 2001) are widely used echo experiments. These can monitor the interaction between electron spin and nucleus in their environment (Deligiannakis *et al.*, 2000). Furthermore, in quantum computation and spintronics, spin is used to store information, opening a new research direction in pulse EPR.

Double Electron-Electron Resonance (DEER) is one of the most popular pulse EPR experiments, also known as *Pulse Electron Double Resonance (PELDOR)* (Schweiger and Jeschke, 2001). Two different frequencies are used in DEER to control different spins in order to exploit their coupling strengths, most importantly the dipolar coupling, which depends on the inverse cube of the distance between spins. Thus, the distance between them can be inferred from the dipolar spin coupling strength as measured from the spacing of the *Pake dou*blets in the Fourier transform of the time-domain DEER signal; for more detail see Misra (2011). Thus, DEER is most commonly used to measure distances in biological systems, which can be used to study the structure of large biological molecules.

There are available two very general open-source software libraries for simulating pulse EPR signals: SPINACH (Hogben *et al.*, 2011) and SPIDYAN (Pribitzer *et al.*, 2016).

3. LVN equation in pulse EPR

3.1 Liouville von Neuman Equation

The evolution of the density matrix, ρ, taking into account relaxation effects is expressed in Liouville space as described, e.g. by Abragam (1961), Jeener (1982), Redfield (1957), Gamliel and Levanon (1995), Misra and Li (2018):

$$\frac{d}{dt}\rho(\Omega,t) = -i\left[\widehat{H},\rho(t)\right] - \hat{\Gamma}(\rho(t)-\rho_0) \tag{3.1}$$

In Eq.(3.1), $\widehat{H} = \widehat{H}_0 + \widehat{H}_1$ is the Hamiltonian operator, where $\widehat{H}_0$ and $\widehat{H}_1$ are its time-independent and time-dependent parts, respectively; $\hat{\Gamma}$ is the relaxation operator, and assumed to be time independent here. (Throughout this thesis, the single and double carets “$\widehat{\ }$” and “$\widehat{\widehat{\ }}$” will be used to denote the operator and the superoperator, respectively.) In Eq. (3.1) ρ_0 is the initial thermal equilibrium density matrix, governed by the Boltzmann distribution:

$$\rho_0 = \frac{\left[\exp(-\hbar\widehat{H}/kT)\right]}{Tr\left[\exp(-\hbar\widehat{H}/kT)\right]} \propto S_z \tag{3.2}$$

Equation (2.1) is an operator equation, which can be expressed as a matrix equation in a given set of operators $|i\rangle\langle j|; i,j = 1,2,\ldots.,n$, where $|i\rangle$ are the eigenvectors of $\widehat{H}$. The coefficients, ρ_{ij}, H_{ij} in the expansion of the operators ρ and $\widehat{H}$ in this basis, respectively, are then used to write the corresponding matrix equation.

The difference between the time-dependent density matrix and the equilibrium density matrix ρ_0 is denoted as $\chi(t) \equiv \rho(t) - \rho_0$. In view of Eq. (3.2) ρ_0 commutes with $\widehat{H}$. Then Eq. (3.1) can be expressed as follows:

$$\frac{d}{dt}\chi(t) = -i\left[\widehat{H},\chi(t)\right] - \hat{\Gamma}\chi(t) \tag{3.3}$$

LVN is a matrix differential equation.

LVN has problems in Hilbert space, since LVN is a matrix differential equation, where each of $\hat{\rho}$, $\hat{\rho}_0$, $\widehat{H}$ is a 2-dimensional matrix, but $\hat{\Gamma}$, the relaxation operator, is a four-index tensor, because it is a tensor that connects two elements of the density matrix, which are characterized by two indices each; see Eq. (3.10) below. Hence it cannot be directly solved in Hilbert space. Need to formulate it into Liouville space.

One can use a mathematical formalism, i.e. vectorization of a matrix A to transform LVN equation from Hilbert to Liouville space. Let $a_i \in \mathbb{C}^m, i = 1,2,\cdots,n,$ be column vectors of dimension m, and $A \in \mathbb{C}^{m\times n}$ be a matrix of dimension $m \times n$. The columns of matrix A consist of the column vectors, i.e. $A = [a_1, a_2, \cdots, a_n]$. The vector operator $Col(A)$ is defined by Magnus and Neudecker (1995).

$$Col(A) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \in \mathbb{C}^{mn} \tag{3.4}$$

Supposing the density operator ρ, the Hamiltonian operator H and the relaxation operator Γ are matrices of dimension $n \times n$, then by vectorizing the operator equation (3.3) in terms of its matrix elements in a chosen representation, one obtains

$$\frac{d}{dt} Col(\chi(t)) = -iCol([\hat{H}, \chi(t)]) - Col\{\hat{\Gamma}\chi(t)\} \tag{3.5}$$

and on the right side

$$\begin{aligned} &Col([\hat{H}, \chi]) \\ &= Col(\hat{H}\chi - \chi\hat{H}) \\ &= (I_n \otimes \hat{H} - \hat{H}^T \otimes I_n) Col(\chi) \\ &= \hat{\hat{H}}\hat{\hat{\chi}} \end{aligned} \tag{3.6}$$

where I_n is unit matrix of dimension $n \times n$, and

$$\hat{\hat{H}} = (I_n \otimes \hat{H} - \hat{H}^T \otimes I_n) \tag{3.7}$$

is called the Hamiltonian superoperator in Liouville space with dimension $n^2 \times n^2$. $\hat{H}^T$ denotes the matrix transpose.

and $\hat{\hat{\chi}}$ is the column vector of dimension $n^2 \times 1$

$$\hat{\hat{\chi}} = Col(\chi)$$

For matrix $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{p \times q}$, operator ⊗ is the Kronecker Product defined by:

$$C = A \otimes B \tag{3.8}$$

$$c_{ij,kl} = a_{i,k} b_{j,l} \tag{3.9}$$

Vectorizing the relaxation component on the right side of Eq.(3.5):

$$Col\{\hat{\Gamma}\chi\} = \hat{\hat{\mathrm{R}}}\hat{\hat{\chi}} \tag{3.10}$$

$\hat{\hat{\mathrm{R}}}$ is is the matrix representing the relaxation superoperator in Liouville space with dimension $n^2 \times n^2$. Its elements are defined as:

$$\hat{\hat{R}}_{\alpha \times n+\beta, \alpha' \times n+\beta'} = \hat{\Gamma}_{\alpha\beta\alpha'\beta'} \tag{3.11}$$

In Eq. (3.11), $\alpha, \beta, \alpha', \beta' = 1,2, \ldots., n$.

Combining all the vectorized density matrix, Hamiltonian and relaxation components, LVN is transformed into Liouville space as:

$$\begin{aligned}\frac{d\hat{\hat{\chi}}}{dt} &= -i\hat{\hat{H}}\hat{\hat{\chi}} - \hat{\hat{R}}\hat{\hat{\chi}} \\ &= -\hat{\hat{L}}\hat{\hat{\chi}}\end{aligned} \tag{3.12}$$

where the Liouvillian

$$\hat{\hat{L}} = i\hat{\hat{H}} + \hat{\hat{R}} \tag{3.13}$$

$\hat{\hat{L}}$ is called the Liouville superoperator.

Equation (3.12) is the key equation to solve using Liouville space, taking into account relaxation effects.

If $\hat{\hat{H}}$ and $\hat{\hat{R}}$ are independent of time, then

$$\hat{\hat{\chi}}(t) = e^{-(t-t_0)\hat{\hat{L}}}\hat{\hat{\chi}}(t_0) \tag{3.14}$$

$\hat{\hat{\chi}}(t_0)$ is the density matrix at the beginning time of the evolution, t_0.

And

$$\hat{\hat{\rho}}(t) = e^{-(t-t_0)\hat{\hat{L}}}\left(\hat{\hat{\rho}}(t_0) - \hat{\hat{\rho}}_0\right) + \hat{\hat{\rho}}_0 \tag{3.15}$$

3.2 Rotating frame

The calculations are carried in the rotating frame during a pulse, wherein the spin is in resonance, so that the effective magnetic field $B_{eff} = \left(B - \frac{\hbar\omega}{g\mu_B}\right)$ becomes zero.

3.3 Solution of LVN during free evolution

Free evolution occurs during the period following the application of a pulse, and the density matrix of the spin system is only affected by the Hamiltonian operator $\hat{H}_0$ and the relaxation operator $\hat{\Gamma}$, both of which are independent of time. The LVN equation can then be solved using the following procedure.

The Hamiltonian superoperator is from Eq. (3.7)

$$\hat{\hat{H}} = I_n \otimes \hat{H}_0 - \hat{H}_0^T \otimes I_n \tag{3.16}$$

Let

$$\boldsymbol{\rho} = \boldsymbol{\sigma} + \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0 \tag{3.17}$$

Using the expression eq. (3.17) for the LVN equation, one obtains

$$\frac{d}{dt}\left(\boldsymbol{\sigma} + \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\right) = -\hat{\hat{L}}\left(\boldsymbol{\sigma} + \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\right) + \hat{\hat{R}}\boldsymbol{\rho}_0 \tag{3.18}$$

$$\frac{d\boldsymbol{\sigma}}{dt} + \frac{d}{dt}\left(\hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\right) = -\hat{\hat{L}}\boldsymbol{\sigma} - \hat{\hat{L}}\hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0 + \hat{\hat{R}}\boldsymbol{\rho}_0 \tag{3.19}$$

Because $\hat{\hat{L}}, \hat{\hat{R}}\ and\ \boldsymbol{\rho}_0$ are independent of time

$$\frac{d}{dt}\left(\hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\right) = 0 \tag{3.20}$$

and the last two terms on the right side in Eq. (3.19) cancel.

one obtains

$$\frac{d\boldsymbol{\sigma}}{dt} = -\hat{\hat{L}}\boldsymbol{\sigma} \tag{3.21}$$

Whose solution is

$$\boldsymbol{\sigma}(t) = e^{-(t-t_0)\hat{\hat{L}}}\boldsymbol{\sigma}(t_0) \tag{3.22}$$

Using Eq. (3.17), one obtains

$$\boldsymbol{\rho}(t) = e^{-(t-t_0)\hat{\hat{L}}}\left(\boldsymbol{\rho}(t_0) - \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\right) + \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0 \tag{3.23}$$

If the matrix $\hat{\hat{L}}$ can be diagonalized, that is,

$$P^{-1}\hat{\hat{L}}P = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} = \hat{\hat{D}} \tag{3.24}$$

or

$$\hat{\hat{L}} = P\hat{\hat{D}}P^{-1} \tag{3.25}$$

Where P is the matrix whose column vectors are the eigenvectors of $\hat{\hat{L}}$. That is, writing P as a block matrix of its column vectors $\vec{\alpha}_i$

$$P = (\vec{\alpha}_1 \quad \vec{\alpha}_2 \quad \ldots \quad \vec{\alpha}_n) \tag{3.26}$$

or

$$\hat{\hat{L}}\vec{\alpha}_i = \lambda_i \vec{\alpha}_i \quad (i = 1,2,\cdots,n) \tag{3.27}$$

Or

$$\hat{\hat{L}}\boldsymbol{\sigma} = P\hat{\hat{D}}P^{-1}\boldsymbol{\sigma} \tag{3.28}$$

The Eq. (3.21) can now be expressed as

$$\frac{d\boldsymbol{\sigma}}{dt} = -P\hat{\hat{D}}P^{-1}\boldsymbol{\sigma} \tag{3.29}$$

Multiplying this from the left by the time independent operator P^{-1}, where P is defined in Eq. (3.26), one obtains

$$\frac{dP^{-1}\boldsymbol{\sigma}}{dt} = -\hat{\hat{D}}P^{-1}\boldsymbol{\sigma} \tag{3.30}$$

Define

$$\boldsymbol{\sigma}' \equiv P^{-1}\boldsymbol{\sigma} \tag{3.31}$$

One obtains for Eq. (3.30)

$$\frac{d\boldsymbol{\sigma}'}{dt} = -\hat{\hat{D}}\boldsymbol{\sigma}' \tag{3.32}$$

the solution of which is

$$\begin{aligned}\boldsymbol{\sigma}'(t) &= e^{\hat{\hat{D}}(t-t_0)}\boldsymbol{\sigma}'(t_0)\\ &= \begin{bmatrix} e^{-\lambda_1\tau} & 0 & 0 & 0 \\ 0 & e^{-\lambda_2\tau} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{-\lambda_n\tau} \end{bmatrix}\boldsymbol{\sigma}'(t_0)\end{aligned} \tag{3.33}$$

where $\tau = t - t_0$. Using Eq. (3.31), one obtains

$$P^{-1}\boldsymbol{\sigma}(t) = \begin{bmatrix} e^{-\lambda_1\tau} & 0 & 0 & 0 \\ 0 & e^{-\lambda_2\tau} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{-\lambda_n\tau} \end{bmatrix} P^{-1}\boldsymbol{\sigma}(t_0) \tag{3.34}$$

Or multiplying both sides by P, one obtains

$$\boldsymbol{\sigma}(t) = P\begin{bmatrix} e^{-\lambda_1\tau} & 0 & 0 & 0 \\ 0 & e^{-\lambda_2\tau} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{-\lambda_n\tau} \end{bmatrix} P^{-1}\boldsymbol{\sigma}(t_0) \tag{3.35}$$

Using Eqs. (3.17) and (3.35), one obtains

$$\begin{aligned}&\boldsymbol{\rho}(t) - \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0 \\ &= P\begin{bmatrix} e^{-\lambda_1\tau} & 0 & 0 & 0 \\ 0 & e^{-\lambda_2\tau} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{-\lambda_n\tau} \end{bmatrix} P^{-1}\left(\boldsymbol{\rho}(t_0) - \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\right)\end{aligned} \tag{3.36}$$

$$\boldsymbol{\rho}(t) = P\begin{bmatrix} e^{-\lambda_1\tau} & 0 & 0 & 0 \\ 0 & e^{-\lambda_2\tau} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{-\lambda_n\tau} \end{bmatrix} P^{-1}\left(\boldsymbol{\rho}(t_0) - \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\right) + \hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0 \tag{3.37}$$

$$\begin{aligned}\boldsymbol{\rho}(t) = P&\begin{bmatrix} e^{-\lambda_1\tau} & 0 & 0 & 0 \\ 0 & e^{-\lambda_2\tau} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{-\lambda_n\tau} \end{bmatrix} P^{-1}\boldsymbol{\rho}(t_0) \\ &- \left(P\begin{bmatrix} e^{-\lambda_1\tau} & 0 & 0 & 0 \\ 0 & e^{-\lambda_2\tau} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & e^{-\lambda_n\tau} \end{bmatrix} P^{-1} - \boldsymbol{I}\right)\hat{\hat{L}}^{-1}\hat{\hat{R}}\boldsymbol{\rho}_0\end{aligned} \tag{3.38}$$

Where relaxation effects are contained in Liouville superoperator $\hat{\hat{L}}$.

3.4 Solution of LVN during a pulse

During a pulse,

$$\widehat{H}_1 = \hat{\varepsilon}(t) \tag{3.39}$$

where $\hat{\varepsilon}(t)$ is the irradiating microwave pulse with the intensity $B_1 = \omega_1/\gamma_e$ and phase ϕ. This interaction of the B_1 field with the electronic spin is expressed in the rotating frame (see Sec. 3.2 above) as

$$\hat{\varepsilon}(t) = B_1\gamma_e\big(S_x cos\phi + S_y sin\phi\big)I_n = \frac{\omega_1}{2}\big(e^{-i\phi}S_+ + e^{i\phi}S_-\big)I_n \tag{3.40}$$

Where $I_n = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is the identify operator in nuclear space

And here $\hat{\varepsilon}(t) \gg \widehat{H}_0, \hat{\Gamma}$, as in the rotating frame the effective magnetic field is zero, so that

$$\widehat{H} = \widehat{H}_0 + \hat{\Gamma} + \hat{\varepsilon}(t) \approx \hat{\varepsilon}(t) \tag{3.41}$$

The LVN becomes

$$\frac{d}{dt}\rho = -i[\hat{\varepsilon}(t), \rho] \tag{3.42}$$

The solution of Eq. (3.42) is Mims (1965)

$$\rho\left(t_0+t_p\right)=exp\left(-i\hat{\varepsilon}t_p\right)\rho(t_0)exp\left(i\hat{\varepsilon}t_p\right)=\hat{\hat{P}}\left(t_0+t_p\right)\rho(t_0) \tag{3.43}$$

Where

$\rho\left(t_0+t_p\right)$ is the density matrix after the application of a pulse.

$exp\left(i\hat{\varepsilon}t_p\right)$ is call right propagator and

$exp\left(-i\hat{\varepsilon}t_p\right)$ is called left propagator.

Using the vectorization of matrix products:

$$Col(ABC)=(C^T\otimes A)Col(B) \tag{3.44}$$

One obtains

$$\begin{aligned}&Col\left(\rho\left(t_0+t_p\right)\right)\\&=Col\left(exp\left(-i\hat{\varepsilon}t_p\right)\rho(t_0)exp\left(i\hat{\varepsilon}t_p\right)\right)\\&=\left(\left(exp\left(i\hat{\varepsilon}t_p\right)\right)^T\otimes exp\left(-i\hat{\varepsilon}t_p\right)\right)Col\left(\rho(t_0)\right)\end{aligned} \tag{3.45}$$

So that

$$\boldsymbol{\rho}\left(t_0+t_p\right)=\hat{\hat{P}}\boldsymbol{\rho}\left(t_p\right) \tag{3.46}$$

where

$$\boldsymbol{\rho}(t)=Col\left(\rho(t)\right) \tag{3.47}$$

and

$$\begin{aligned}\hat{\hat{P}}&=\left(exp\left(i\hat{\varepsilon}t_p\right)\right)^T\otimes exp\left(-i\hat{\varepsilon}t_p\right)\\&=\left(exp\left(-i\hat{\varepsilon}t_p\right)\right)^\dagger\otimes exp\left(-i\hat{\varepsilon}t_p\right)\end{aligned} \tag{3.48}$$

$\hat{\hat{P}}$ is called pulse superoperator in Liouville space.

The specific example, in the $|S_zI_z\rangle$ basis of a spin system with $S_z=\pm 1/2\ and\ I_z=\pm 1/2$, and $\hat{\varepsilon}(t)$is given by Eq. (3.40).

Because

$$\mathrm{S}_+I_n|+\pm\rangle=0 \tag{3.49}$$

$$\mathrm{S}_+I_n|-\pm\rangle=|+\pm\rangle \tag{3.50}$$

$$\mathrm{S}_{-}I_n|+\pm\rangle = |-\pm\rangle \tag{3.51}$$

$$\mathrm{S}_{-}I_n|-\pm\rangle = 0 \tag{3.52}$$

One obtains

$$\hat{\varepsilon}(t)|+\pm\rangle = \frac{\omega_1}{2}\left(e^{-i\phi}S_{+} + e^{i\phi}S_{-}\right)|+\pm\rangle = \frac{\omega_1}{2}e^{i\phi}|-\pm\rangle \tag{3.53}$$

$$\hat{\varepsilon}(t)|-\pm\rangle = \frac{\omega_1}{2}\left(e^{-i\phi}S_{+} + e^{i\phi}S_{-}\right)|-\pm\rangle = \frac{\omega_1}{2}e^{-i\phi}|+\pm\rangle \tag{3.54}$$

Thus

$$[\hat{\varepsilon}] = \begin{array}{c} \\ |1\rangle = |++\rangle \\ |2\rangle = |+-\rangle \\ |3\rangle = |-+\rangle \\ |4\rangle = |--\rangle \end{array} \begin{array}{c} \begin{array}{cccc} |1\rangle = |++\rangle & |2\rangle = |+-\rangle & |3\rangle = |-+\rangle & |4\rangle = |--\rangle \end{array} \\ \begin{bmatrix} 0 & 0 & \frac{\omega_1}{2}e^{-i\phi} & 0 \\ 0 & 0 & 0 & \frac{\omega_1}{2}e^{-i\phi} \\ \frac{\omega_1}{2}e^{i\phi} & 0 & 0 & 0 \\ 0 & \frac{\omega_1}{2}e^{i\phi} & 0 & 0 \end{bmatrix} \end{array} \tag{3.55}$$

or

$$[\hat{\varepsilon}] = \frac{\omega_1}{2}\begin{bmatrix} \mathbf{0}_{\boldsymbol{n}} & e^{-i\phi}\boldsymbol{I}_{\boldsymbol{n}} \\ e^{i\phi}\boldsymbol{I}_{\boldsymbol{n}} & \mathbf{0}_{\boldsymbol{n}} \end{bmatrix} = \frac{\omega_1}{2}\Sigma_n \tag{3.56}$$

where

$$\Sigma_z = \begin{bmatrix} \mathbf{0}_{\mathbf{2}} & e^{-i\phi}\boldsymbol{I}_{\boldsymbol{n}} \\ e^{i\phi}\boldsymbol{I}_{\boldsymbol{n}} & \mathbf{0}_{\boldsymbol{n}} \end{bmatrix} \tag{3.57}$$

We also have

$$\begin{aligned} [\hat{\varepsilon}][\hat{\varepsilon}] &= \left(\frac{\omega_1}{2}\right)^2 \begin{bmatrix} \mathbf{0}_{\boldsymbol{n}} & e^{-i\phi}\boldsymbol{I}_{\boldsymbol{n}} \\ e^{i\phi}\boldsymbol{I}_{\boldsymbol{n}} & \mathbf{0}_{\boldsymbol{n}} \end{bmatrix}\begin{bmatrix} \mathbf{0}_{\boldsymbol{n}} & e^{-i\phi}\boldsymbol{I}_{\boldsymbol{n}} \\ e^{i\phi}\boldsymbol{I}_{\boldsymbol{n}} & \mathbf{0}_{\boldsymbol{n}} \end{bmatrix} \\ &= \left(\frac{\omega_1}{2}\right)^2 \begin{bmatrix} \boldsymbol{I}_{\boldsymbol{n}} & \mathbf{0}_{\boldsymbol{n}} \\ \mathbf{0}_{\boldsymbol{n}} & \boldsymbol{I}_{\boldsymbol{n}} \end{bmatrix} \\ &= \left(\frac{\omega_1}{2}\right)^2 \boldsymbol{I}_{\mathbf{4}} \end{aligned} \tag{3.58}$$

The right propagator and left propagator matrix in Hilbert space are expanded as following:

$$\begin{aligned}
[exp(i\hat{\varepsilon}t)] &= \sum_{n=0}^{\infty}\frac{1}{n!}(i\hat{\varepsilon}t)^{n} \\
&= \sum_{n=0}^{\infty}\frac{1}{(2n)!}(i\hat{\varepsilon}t)^{2n} + \sum_{n=0}^{\infty}\frac{1}{(2n+1)!}(i\hat{\varepsilon}t)^{2n+1} \\
&= \sum_{n=0}^{\infty}\frac{(-1)^{n}}{(2n)}(\omega_1 t/2)^{2n}\boldsymbol{I_4} + i\sum_{n=0}^{\infty}\frac{(-1)^{n}}{(2n+1)}(\omega_1 t/2)^{2n+1}\Sigma_z \\
&= cos(\omega_1 t/2)\boldsymbol{I_4} + isin(\omega_1 t/2)\Sigma_z \\
&= cos(\omega_1 t/2)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + isin(\omega_1 t/2)\begin{bmatrix} 0 & 0 & e^{-i\phi} & 0 \\ 0 & 0 & 0 & e^{-i\phi} \\ e^{i\phi} & 0 & 0 & 0 \\ 0 & e^{i\phi} & 0 & 0 \end{bmatrix}
\end{aligned} \tag{3.59}$$

$$[exp(i\hat{\varepsilon}t)]$$

$$= \begin{array}{c|cccc}
 & |1\rangle = |++\rangle & |2\rangle = |+-\rangle & |3\rangle = |-+\rangle & |4\rangle = |--\rangle \\
\hline
|1\rangle = |++\rangle & \cos\frac{\omega_1 t}{2} & 0 & ie^{-i\phi}\sin\frac{\omega_1 t}{2} & 0 \\
|2\rangle = |+-\rangle & 0 & \cos\frac{\omega_1 t}{2} & 0 & ie^{-i\phi}\sin\frac{\omega_1 t}{2} \\
|3\rangle = |-+\rangle & ie^{i\phi}\sin\frac{\omega_1 t}{2} & 0 & \cos\frac{\omega_1 t}{2} & 0 \\
|4\rangle = |--\rangle & 0 & ie^{i\phi}\sin\frac{\omega_1 t}{2} & 0 & \cos\frac{\omega_1 t}{2}
\end{array} \tag{3.60}$$

$$= \begin{array}{c|cccc}
 & |1\rangle = |++\rangle & |2\rangle = |+-\rangle & |3\rangle = |-+\rangle & |4\rangle = |--\rangle \\
\hline
|1\rangle = |++\rangle & \cos\frac{\theta}{2} & 0 & ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\
|2\rangle = |+-\rangle & 0 & \cos\frac{\theta}{2} & 0 & ie^{-i\phi}\sin\frac{\theta}{2} \\
|3\rangle = |-+\rangle & ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\
|4\rangle = |--\rangle & 0 & ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2}
\end{array} \tag{3.61}$$

Finally one obtains

$$[exp(i\hat{\varepsilon}t)] = \begin{bmatrix} \cos\frac{\theta}{2}\boldsymbol{I_n} & ie^{-i\phi}\sin\frac{\theta}{2}\boldsymbol{I_n} \\ ie^{i\phi}\sin\frac{\theta}{2}\boldsymbol{I_n} & \cos\frac{\theta}{2}\boldsymbol{I_n} \end{bmatrix} \tag{3.62}$$

where

$$\boldsymbol{I_n} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.63}$$

Tip angle: $\theta = \omega_1 t$, t is pulse duration.

In same way

$$[exp(-i\hat{\varepsilon}t)]$$

$$= \begin{array}{ccccc} & |1\rangle = |++\rangle & |2\rangle = |+-\rangle & |3\rangle = |-+\rangle & |4\rangle = |--\rangle \\ |1\rangle = |++\rangle & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ |2\rangle = |+-\rangle & 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ |3\rangle = |-+\rangle & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ |4\rangle = |--\rangle & 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{array} \qquad (3.64)$$

$$[exp(-i\hat{\varepsilon}t)] = \begin{bmatrix} \cos\frac{\theta}{2}\boldsymbol{I_n} & -ie^{-i\phi}\sin\frac{\theta}{2}\boldsymbol{I_n} \\ -ie^{i\phi}\sin\frac{\theta}{2}\boldsymbol{I_n} & \cos\frac{\theta}{2}\boldsymbol{I_n} \end{bmatrix} \qquad (3.65)$$

$$\hat{\hat{P}} = \left(exp\left(i\hat{\varepsilon}t_p\right)\right)^T \otimes exp\left(-i\hat{\varepsilon}t_p\right)$$

$$= \begin{bmatrix} \cos\frac{\theta}{2} & 0 & ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & ie^{-i\phi}\sin\frac{\theta}{2} \\ ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix}^T \otimes \begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} \qquad (3.66)$$

$$=$$

$$\begin{bmatrix} \cos\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} & ie^{i\phi}\sin\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} \\ \mathbf{0_4} & & \mathbf{0_4} & \\ ie^{-i\phi}\sin\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} & \cos\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} \\ \mathbf{0_4} & & \mathbf{0_4} & \end{bmatrix}$$

$$+ \begin{bmatrix} & \mathbf{0_4} & & \mathbf{0_4} \\ \mathbf{0_4} & \cos\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} & ie^{i\phi}\sin\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} \\ & \mathbf{0_4} & & \mathbf{0_4} \\ \mathbf{0_4} & ie^{-i\phi}\sin\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} & \cos\frac{\theta}{2}\begin{bmatrix} \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} & 0 & -ie^{-i\phi}\sin\frac{\theta}{2} \\ -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} & 0 \\ 0 & -ie^{i\phi}\sin\frac{\theta}{2} & 0 & \cos\frac{\theta}{2} \end{bmatrix} \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} \begin{bmatrix} \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta \\ -e^{i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} & 0 \\ 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} & \begin{bmatrix} e^{i\phi}\frac{i}{2}\sin\theta & 0 & \sin^2\frac{\theta}{2} & 0 \\ 0 & e^{i\phi}\frac{i}{2}\sin\theta & 0 & \sin^2\frac{\theta}{2} \\ e^{i2\phi}\sin^2\frac{\theta}{2} & 0 & e^{i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & e^{i2\phi}\sin^2\frac{\theta}{2} & 0 & e^{i\phi}\frac{i}{2}\sin\theta \end{bmatrix} & \mathbf{0_4} \\ \mathbf{0_4} & & \mathbf{0_4} & \\ \begin{bmatrix} e^{-i\phi}\frac{i}{2}\sin\theta & 0 & e^{-i2\phi}\sin^2\frac{\theta}{2} & 0 \\ 0 & e^{-i\phi}\frac{i}{2}\sin\theta & 0 & e^{-i2\phi}\sin^2\frac{\theta}{2} \\ \sin^2\frac{\theta}{2} & 0 & e^{-i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & \sin^2\frac{\theta}{2} & 0 & e^{-i\phi}\frac{i}{2}\sin\theta \end{bmatrix} & \mathbf{0_4} & \begin{bmatrix} \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta \\ -e^{i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} & 0 \\ 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} \\ \mathbf{0_4} & & \mathbf{0_4} & \end{bmatrix}$$

$$+\begin{bmatrix} & \mathbf{0_4} & & \mathbf{0_4} \\ \mathbf{0_4} & \begin{bmatrix} \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta \\ -e^{i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} & 0 \\ 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} \end{bmatrix} & \mathbf{0_4} & \begin{bmatrix} e^{i\phi}\frac{i}{2}\sin\theta & 0 & \sin^2\frac{\theta}{2} & 0 \\ 0 & e^{i\phi}\frac{i}{2}\sin\theta & 0 & \sin^2\frac{\theta}{2} \\ e^{i2\phi}\sin^2\frac{\theta}{2} & 0 & e^{i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & e^{i2\phi}\sin^2\frac{\theta}{2} & 0 & e^{i\phi}\frac{i}{2}\sin\theta \end{bmatrix} \\ & \mathbf{0_4} & & \mathbf{0_4} \\ \mathbf{0_4} & \begin{bmatrix} e^{-i\phi}\frac{i}{2}\sin\theta & 0 & e^{-i2\phi}\sin^2\frac{\theta}{2} & 0 \\ 0 & e^{-i\phi}\frac{i}{2}\sin\theta & 0 & e^{-i2\phi}\sin^2\frac{\theta}{2} \\ \sin^2\frac{\theta}{2} & 0 & e^{-i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & \sin^2\frac{\theta}{2} & 0 & e^{-i\phi}\frac{i}{2}\sin\theta \end{bmatrix} & \mathbf{0_4} & \begin{bmatrix} \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 \\ 0 & \cos^2\frac{\theta}{2} & 0 & -e^{-i\phi}\frac{i}{2}\sin\theta \\ -e^{i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} & 0 \\ 0 & -e^{-i\phi}\frac{i}{2}\sin\theta & 0 & \cos^2\frac{\theta}{2} \end{bmatrix} \end{bmatrix}$$

Where

$$\mathbf{0_4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.67}$$

3.5 Phenomenological relaxation matrix elements

In Liouville space spanned by the $\widehat{\mathrm{H}}_0$basis set, the diagonal and the off-diagonal parts of the relaxation matrix are described by Freed and Fraenkel (1963), and Freed (1965, 1965a, 1968):

$$\widehat{\mathrm{R}}_{\alpha\beta,\alpha\beta} = \widehat{\mathrm{R}}_{\beta\alpha,\beta\alpha} = -\left(\frac{1}{T_2}\right)_{\alpha\beta} \tag{3.68}$$

$$\widehat{\mathrm{R}}_{\alpha\alpha,\beta\beta} = W_{\alpha\beta} \tag{3.69}$$

$$\hat{\mathrm{R}}_{\alpha\alpha,\alpha\alpha} = -\sum\nolimits_{\gamma\neq\alpha} W_{\alpha\gamma} \qquad (3.70)$$

Otherwise

$$\hat{\mathrm{R}}_{\alpha\beta,\zeta\eta} = 0 \qquad (3.71)$$

where $\alpha,\ \alpha',\ \beta,\ \beta'$ designate the eigenstates of the Hamiltonian $\hat{H}_0$, T_2 is the spin-spin relaxation time,
$W_{\alpha\beta}$ is the transition probability from state α to state β, which leads to spin relaxation.
Exchange and dipolar relaxation Freed (1979)

$$W^{HE}_{\alpha^+\beta^+} = -W^{HE}_{\alpha^-\beta^+} = \frac{\omega_{HE}}{2N} = \frac{\omega_{HE}}{4} \qquad (3.72)$$

$$W^{HE}_{\alpha^\pm\beta^-} = 0 \qquad (3.73)$$

$$W^{HE}_{\alpha_j^\pm\alpha_j^\mp} = \frac{\omega_H}{2} \qquad (3.74)$$

$$W^{HE}_{\alpha_j^+\gamma_k^+} = \frac{\omega_H}{2} = -W^{HE}_{\alpha_j^-\gamma_k^+} = \frac{2\omega_H}{N}, \qquad all\,\gamma_k^+ \qquad (3.75)$$

$$W^{HE}_{\alpha_j^\pm\gamma_k^-} = 0, \qquad all\,\gamma_k^- \qquad (3.76)$$

3.6 Coherence pathways

The basic details of coherence pathways have been described earlier in Sec. 2.3, especially with regard to the transitions between the energy levels corresponding to the various electronic magnetic quantum numbers. A coherence pathway is the sequence of coherence orders that the magnetization evolves through during a pulse sequence. All experiments start with zero order coherence (z-magnetization) and should end with a coherence order of -1, which is by convention the one that is detected by the quadrature detector. Without quadrature detection the +1 coherences would be equally detectable, all higher orders are not correlated with observable magnetization.

The selection of the appropriate coherence pathways can be experimentally achieved by phase cycling, or by the use of pulse field gradients e.g. Bain (1984), Bodenhausen *et al.* (1984) Gemperle *et al.* (1990). The properties of the RF-pulses, e.g. flip angle, offset effects, inhomogeneity, require a weighting over the different coherence transfer pathways e.g. Misra *et al.* (2009), Håkansson *et al.* (2013), Franck *et al.* (2015). The coherence pathways S_{c-} and S_{c+} for obtaining SECSY and echo-ELDOR signals, used commonly, are depicted below in Figs. 4.4 and 4.7, respectively. In this thesis, the signal is calculated over the coherent pathway S_{c-} in accordance with that used by Lee *et al.* (1993).

Selection of coherent pathways by appropriate selection of the corresponding elements of the density matrix. For a single electron-nuclear coupled system with electron spin S=1/2, nuclear spin I=1/2, the dimension of the coupled spin Hamiltonian is $(2S_1 + 1) \times (2I_1 + 1) = 4$, so that the density matrix in the electron-nuclear spin-coupled direct product space the toal density matrix $\rho = \rho_e \otimes \rho_n$, where ρ_e and ρ_n are, respectively, the density matrices in the electronic and nuclear subspaces, In the following, the coherence order is defined as follows; here p is the coherence order, which represents transverse magnetization, corresponding to spins rotating in a plane perpendicular to the external field, e.g. Misra and Freed (2011), Stoll and Kasumaj (2008), Misra and Li (2018).

(i) For the pathway with coherence order $p = 1$, all matrix elements of the density matrix, ρ_e , are put equal to zero, except for those corresponding to $\rho_e(1{,}2)$.

(ii) For the pathway with coherence order $p = 0$ all matrix elements of the density matrix, ρ_e, are put equal to zero, except for those corresponding to $\rho_e(1{,}1)$ and $\rho_e(2{,}2)$.

(iii) For the pathway with coherence order $p = -1$, all matrix elements of the density matrix, ρ_e, are put equal to zero, except for those corresponding to $\rho_e(2{,}1)$.

3.7 Simulation procedure

The algorithm procedure for calculation of pulse EPR signal is as follows. The time evolution of the density matrix includes evolution under the action of the static Hamiltonian, H_0 , in the absence of pulses, and under the action of pulse H_1, as illustrated in above.

Calculation of the final density matrix, ρ_f, corresponding to the coherence pathways S_{c-} in Figures 4.4 and 4.7 for the evolution of the density matrix in the absence and presence of pulses, respectively. The Fourier transform (FT) of the two-dimensional (2D) time domain signal $S(t_1, t_2)$, is the corresponding 2D-FT signal, $S(\omega_1, \omega_2)$.

The 2D time-domain signal is calculated from ρ_f as follows:

$$S(t_1, t_2) = Tr(S_+ \rho_f) = Tr\left((S_x + iS_y)\rho_f\right), \tag{3.77}$$

4. Simulation of SECSY and echo-ELDOR

4.1 Spin Hamiltonian and parameters

For the specific case of a single nucleus (I = ½) interacting with an unpaired electron (S = ½) by the hyperfine (HF) interaction tensor, where the HF has the same principal axes as the g matrix, the total Hamiltonian can be expressed as the sum of static ($\widehat{H}_0$) and pulse ($\widehat{H}_1$) Hamiltonians (Lee *et al*., 1993):

$$\widehat{H} = \widehat{H}_0 + \widehat{H}_1 \tag{4.1}$$

where

$$\widehat{H}_0 = CS_z - \omega_n I_z + AS_z I_z + \frac{1}{2}BS_z I_+ + \frac{1}{2}B^* S_z I_- \tag{4.2}$$

where

$$C = \frac{\beta_e\, B_0}{h}\left[\bar{g} + F\frac{1}{2}(3\cos^2\beta - 1) + F^{(2)}\sin^2\beta\cos(2\gamma)\right] \tag{4.3}$$

$$A = -2\pi\left[\bar{a} + D\frac{1}{2}(3\cos^2\beta - 1) + D^{(2)}\sin^2\beta\cos(2\gamma)\right] \tag{4.4}$$

$$B = -4\pi\left\{D\frac{3}{4}\sin\beta\cos\beta - D^{(2)}\frac{1}{2}\sin\beta\,[\cos\beta\cos(2\gamma) - i\,\mathrm{s}\,in(2\gamma)]\right\} \tag{4.5}$$

and

$$\bar{g} = \frac{1}{3}\left(g_{xx} + g_{yy} + g_{zz}\right) \tag{4.6}$$

$\bar{g}$ – isotropic part of g-tensor

$g_{xx},\ g_{yy}, g_{zz}$ are the components of the g-tensor in their principal-axes systems

$$\bar{a} = \frac{1}{3}\left(A_{xx} + A_{yy} + A_{zz}\right) \tag{4.7}$$

$A_{xx},\ A_{yy}, A_{zz}$ a are the components of the hyperfine tensor in their principal-axes systems

$$F = \frac{2}{3}\left(g_{zz} - \frac{1}{2}\left(g_{xx} + g_{yy}\right)\right) \tag{4.8}$$

$$D = \frac{2}{3}\left(A_{zz} - \frac{1}{2}\left(A_{xx} + A_{yy}\right)\right) \tag{4.9}$$

$$F^{(2)} = \frac{1}{2}\left(g_{xx} - g_{yy}\right) \tag{4.10}$$

$$D^{(2)} = \frac{1}{2}\left(A_{xx} - A_{yy}\right) \tag{4.11}$$

ω_n – the nuclear Larmor frequency for the nucleus,

$\Omega(\alpha, \beta, \gamma)$ – the Euler angles which describe the orientation of the g tensor with respect to the static magnetic field,

β – the angle between the symmetry axis (“z-axis”) of the hyperfine tensor and the z-axis of the laboratory system.

Spin $S_z = \pm 1/2\,, I_z = \pm 1/2$

$$S_x = \frac{1}{2}\begin{bmatrix} \mathbf{0} & I_2 \\ I_2 & \mathbf{0} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.12}$$

$$S_y = \frac{1}{2}\begin{bmatrix} \mathbf{0} & -iI_2 \\ iI_2 & \mathbf{0} \end{bmatrix} = \frac{i}{2}\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.13}$$

$$S_z = \frac{1}{2}\begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & -I_2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{4.14}$$

$$\left(S_x + iS_y\right) = \frac{1}{2}\begin{bmatrix} \mathbf{0} & I_2 \\ I_2 & \mathbf{0} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} \mathbf{0} & -I_2 \\ I_2 & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & I_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.15}$$

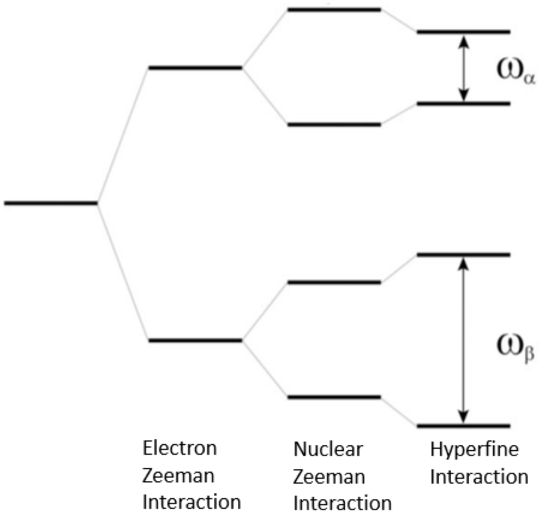


Figure 4.1 Four-level electron-nuclear spin coupled (S=1/2, I=1/2) system

where

$$\omega_\alpha = \sqrt{\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2}$$

$$\omega_\beta = \sqrt{\left(\omega_n + \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2}$$

4.2 Relaxation

The following specific values for the matrix elements, as given by Freed (1979), are used here:

$$\hat{\hat{\mathrm{R}}}_{\alpha\beta,\alpha\beta} = \hat{\hat{\mathrm{R}}}_{\beta\alpha,\beta\alpha} = -\left(\frac{1}{T_2}\right)_{\alpha\beta} \tag{4.16}$$

$$\hat{\hat{\mathrm{R}}}_{\alpha\alpha,\beta\beta} = W_{\alpha\beta} \tag{4.17}$$

$$\hat{\hat{\mathrm{R}}}_{\alpha\alpha,\alpha\alpha} = -\sum\nolimits_{\gamma\neq\alpha} W_{\alpha\gamma} \tag{4.18}$$

Otherwise

$$\widehat{\widehat{\mathrm{R}}}_{\alpha\beta,\zeta\eta} = 0 \tag{4.19}$$

The relaxation pathways among the various eigenstates of the two electrons in the lattice, as defined in the *H₀* basis and described by Freed (1979), are shown in the Figure below where a and b indicate the nuclear sublevels of electronic magnetic quantum number $M_S = 1/2$, whereas c and d indicate the nuclear sublevels of $M_S = -1/2$.

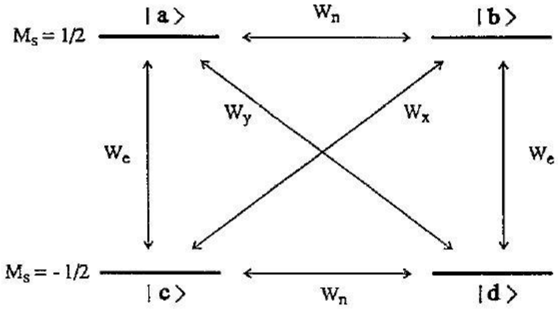


Figure 4.2 Relaxation pathways defined in the H_0 basis. [Reproduced by the permission of the author (Freed, 1979).]

Corresponding to the above diagram, it is assumed that (Lee *et al*. 1993)

$$W_{ab} = W_{ba} = W_{cd} = W_{dc} = W_n \tag{4.20}$$

$$W_{ac} = W_{ca} = W_{bd} = W_{db} = W_e \tag{4.21}$$

$$W_{ad} = W_{da} = W_y \tag{4.22}$$

$$W_{bc} = W_{cb} = W_x \tag{4.23}$$

and

$$(T_2)_{ac} = (T_2)_{bd} = (T_2)_{ad} = (T_2)_{bc} = T_{2e} \tag{4.24}$$

$$(T_2)_{ab} = (T_2)_{cd} = T_{2n} \tag{4.25}$$

The non-zero elements of the relaxation matrix, corresponding to the above diagram, as obtained from Eqs.(4.16) -(4.19)) are (Lee *et al*. 1993):

$$\widehat{\widehat{\mathrm{R}}}_{ab,ab} = \widehat{\widehat{\mathrm{R}}}_{ba,ba} = \widehat{\widehat{\mathrm{R}}}_{cd,cd} = \widehat{\widehat{\mathrm{R}}}_{dc,dc} = -\frac{1}{T_{2n}} \tag{4.26}$$

$$\widehat{\widehat{\mathrm{R}}}_{ac,ac} = \widehat{\widehat{\mathrm{R}}}_{ca,ca} = \widehat{\widehat{\mathrm{R}}}_{ad,ad} = \widehat{\widehat{\mathrm{R}}}_{da,da} = \widehat{\widehat{\mathrm{R}}}_{bc,bc} = \widehat{\widehat{\mathrm{R}}}_{cb,cb} = \widehat{\widehat{\mathrm{R}}}_{bd,bd} = \widehat{\widehat{\mathrm{R}}}_{db,db} = -\frac{1}{T_{2e}} \tag{4.27}$$

$$\hat{\hat{\mathrm{R}}}_{aa,aa} = \hat{\hat{\mathrm{R}}}_{dd,dd} = -W_e - W_n - W_y \tag{4.28}$$

$$\hat{\hat{\mathrm{R}}}_{bb,bb} = \hat{\hat{\mathrm{R}}}_{cc,cc} = -W_e - W_n - W_x \tag{4.29}$$

$$\hat{\hat{\mathrm{R}}}_{aa,bb} = \hat{\hat{\mathrm{R}}}_{bb,aa} = \hat{\hat{\mathrm{R}}}_{cc,dd} = \hat{\hat{\mathrm{R}}}_{dd,cc} = W_n + \frac{\omega_{HE}}{4} \tag{4.30}$$

$$\hat{\hat{\mathrm{R}}}_{aa,cc} = \hat{\hat{\mathrm{R}}}_{bb,dd} = \hat{\hat{\mathrm{R}}}_{cc,aa} = \hat{\hat{\mathrm{R}}}_{dd,bb} = W_e + \frac{\omega_{HE}}{4} \tag{4.31}$$

$$\hat{\hat{\mathrm{R}}}_{bb,cc} = \hat{\hat{\mathrm{R}}}_{cc,bb} = W_x - \frac{\omega_{HE}}{4} \tag{4.32}$$

$$\hat{\hat{\mathrm{R}}}_{aa,cc} = \hat{\hat{\mathrm{R}}}_{dd,aa} = W_y - \frac{\omega_{HE}}{4} \tag{4.33}$$

otherwise $\hat{\hat{\mathrm{R}}}_{\alpha\beta,\zeta\eta} = 0$. Here $\omega_{HE} = 0$ as used by LPF (Lee 1993).

.

4.3 Gaussian inhomogeneous broadening effect

In accordance with LPF (Lee *et al*. 1993), the Gaussian inhomogeneous broadening effect, over and above the relaxation effect, in the frequency-domain along ω_2 $(= 2\pi\nu)$, corresponding to the step time t_2, as depicted in Figs. 3.1, is taken into account by the following time-domain dependence:

$$f_b(t_2) = f(t_2)\frac{1}{\sqrt{2\pi}\Delta}\int_{-\infty}^{\infty} exp\left(-\frac{\nu^2}{2\Delta^2}\right)e^{-i2\pi\nu t_2}d\nu = f(t_2)e^{-2(\pi\Delta t_2)^2} \tag{4.34}$$

where $f_b(t_2)$ is the Gaussian-broadened signal along t$_2$ and Δ is the Gausssian inhomogeneous broadening parameter expressed in frequency units.

4.4 Illustrative Examples

The examples considered for illustration here are the same as those considered in (Lee 1993) for the single-crystal case. The sample used is an irradiated malonic-acid crystal (McConnell 1960), wherein an unpaired electron spin $S = 1/2$ is in interaction with a single nucleus, $I = 1/2$, by hyperfine (HF) interaction, with the principal axes of the hyperfine (HF) tensor $\tilde{A}$ and those of the $\tilde{g}$ matrix are assumed to be coincident. The orientations of the crystalline axes X, Y, Z are shown in Figure 5.3, whose caption describes the angles (θ, ϕ), which are related to the Euler angles $\eta = (\alpha, \beta, \gamma) = (\chi, \theta, \phi)$ used in the transformation of the static spin Hamiltonian to the laboratory frame.

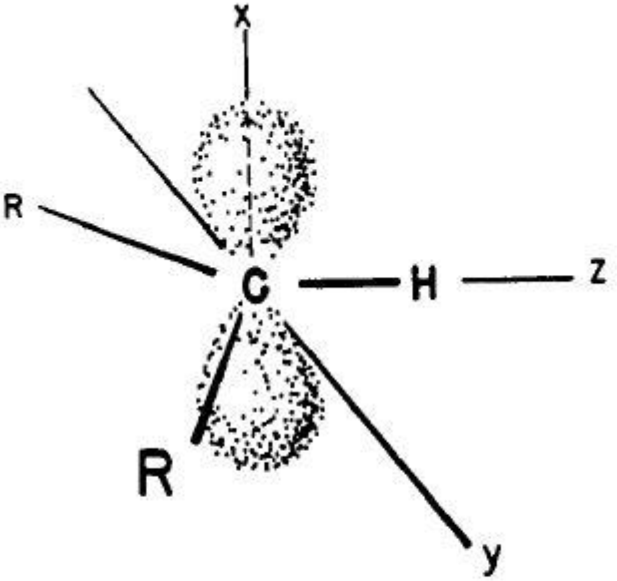


Figure 4.3 Relation of the structure of the malonic acid radical R to the principal axes of the $\tilde{g}$, $\tilde{A}$ matrices [(reproduced by the permission of the authors, Lee *et al*. (1993).]

R = ·CH(COOH)$_2$, to the principal axes (x, y, z) of the hyperfine tensor. Here, the z axis is along the C-H bond direction and the x axis is perpendicular to the plane of the three carbon atoms (McConnell *et al*., 1960). The direction of the external static field B$_0$ is defined by the angles θ and ϕ, where θ is the angle between Bo and the z axis, and ϕ is the angle between the x axis and the projection of B$_0$ on the xy plane.

Electronic Structure of ·CH(COOH)$_2$ (McConnell *et al*., 1960).

Hyperfine interactions in ·CH(COOH)$_2$ indicate that this molecule is a simple π-electron radical with essentially a unit odd-electron spin density on the central carbon atom. The radical is a π-electron radical in the sense that the odd electron spin is largely localized in an orbital antisymmetric to the plane containing three carbon atoms and the hydrogen atom.

Malonic acid crystals were irradiated with 1 Mrad ^{60}Co γ radiation (Lee *et al*., 1993). Soon after irradiation the cw-EPR spectrum showed several hf lines due to different kinds of radicals produced, but in aged samples (a couple of days after irradiation), only two strong lines persist. The strong doublet is due to the hf coupling of the π electron of the α carbon with the α proton of the C-H fragment of the ·CH (COOH)$_2$ radical of malonic acid.

More details of density matrix analysis for nuclear modulation analysis, including the Hamiltonian and basis vectors, are given in Appendix A of Lee *et al*.(1993). In this experiment the various parameters are as follows: the external magnetic field $B_0 = 3000.0\, Gauss$; the $\pi/2$ pulse is of duration $\sim 5ns$; nuclear Zeeman frequency $\omega_n = 14.5MHz$; the spin-Hamiltonian parameters: $\tilde{g} = (g_{xx}, g_{yy}, g_{zz}) = (2.0026, 2.0035, 2.0033)$; $\tilde{A} = (A_{xx}, A_{yy}, A_{zz}) = (-61.0MHz, -91.0MHz, -29.0MHz)$.

The input values used in the simulation of the time-domain signals, as described as follows (Lee *et al.*, 1993): Gaussian inhomogeneous broadening $\Delta = 4MHz$; electron spin–spin relaxation time, $T_{2e} = 0.900\mu s$; nuclear spin–spin relaxation time, $T_{2n} = 22\mu s$; inverse electron spin–spin relaxation time, $W_e = 0.0167\mu s^{-1}$; inverse nuclear spin–spin relaxation time, $W_n = 0.00714\mu s^{-1}$; inverse electron nuclear spin–spin relaxation time, $W_x = 0.00617\mu s^{-1}$; inverse electron-nuclear spin–spin relaxation time $W_x = W_y$; inverse Heisenberg exchange relaxation time, $\omega_{HE} = 0.0\mu s^{-1}$.

4.4.1 SECSY signals

One measures the transfer of coherence from one EPR allowed transition to another one in a SECSY experiment. Here the second pulse refocuses hyperfine and resonance offsets. This allows measurement of the variation of the phase memory time across the EPR spectrum

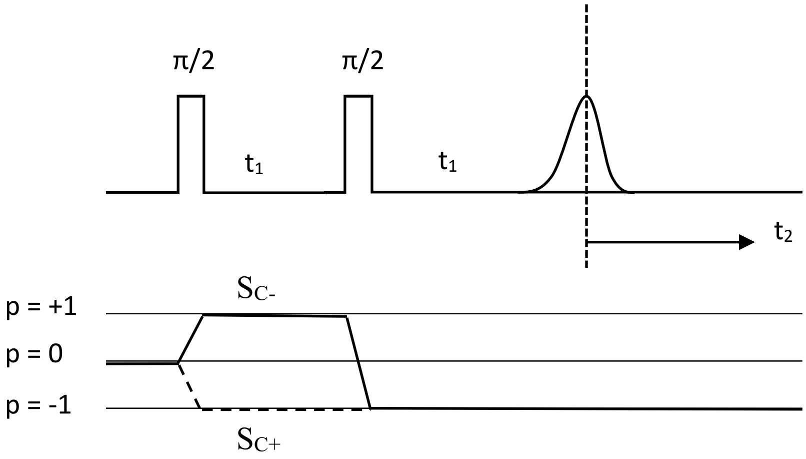


Figure 4.4 Pulse sequence for obtaining SECSY signal.

The t_1 time between the two pulses is stepped and this echo decay is measured as a function of t_2. (Bottom) Coherence pathways used for calculating SECSY signal for an unpaired electron (S = ½) interacting with a single nucleus (I = ½). p is the coherence order, which represents transverse magnetization, corresponding to spins rotating in a plane perpendicular to the external field (Lee *et al.*, 1993).

The SECSY signals were calculated, corresponding to the experiments of LPF (Lee *et al.*, 1993) for: (i) as shown in Figure 4.5, for three orientations of the external magnetic field $(\theta, \phi) = (5^{\circ}, 0^{\circ}), (30, 0^{\circ}), (50^{\circ}, 0^{\circ})$ in the zx-quadrant, so that the corresponding Euler angles are $(\alpha, \beta, \gamma) = (0, -\theta, 0)$; (ii) as shown in Figure 4.6, for two orientations $(\theta, \phi) = (5^{\circ}, 90^{\circ})$ and $(45^{\circ}, 90^{\circ})$ in the zy-quadrant, which correspond to $(\alpha, \beta, \gamma) = (0, \theta, 90^{\circ})$. [The orientation of the external static field B_0 is defined by the angles θ and ϕ, where θ is the angle between B_0 and the z axis, and ϕ is the angle between the x axis and the projection of B_0 on the xy plane.]

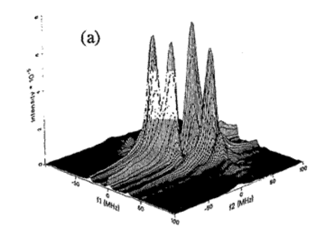
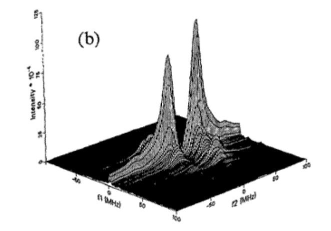
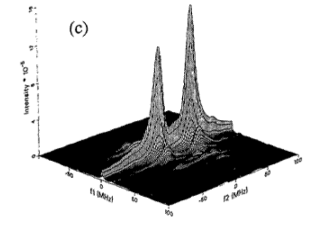
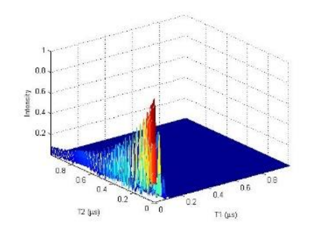
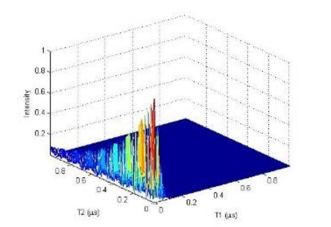
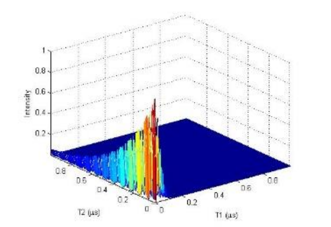
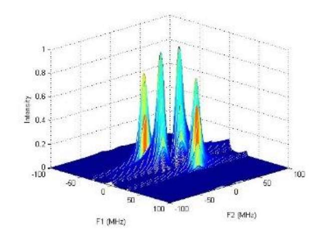
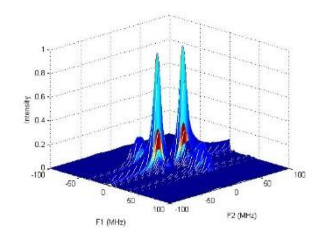
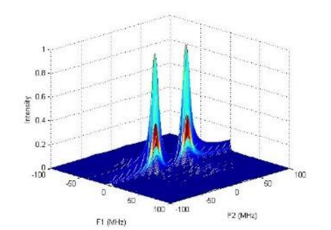
Experiment FT spectra [Lee *et al.* (1993)]			
Simulation: Time-domain signal (this work)			
Simulation – Fourier transform (this work)			

Figure 4.5 Simulated time-domain and Fourier transforms of SECSY signals

The experimental Figures in the top row are reproduced by the permission of the authors from Figure 8 of Lee *et al.* (1993) (upper). Simulations (bottom) of SECSY spectra at (θ, ϕ)orientations of (a) (5°, 0°), (b) (30°, 0°), (c) (50°, 0°) in the zx-quadrant $[(\alpha, \beta, \gamma) = (0, -\theta, 0)]$. The corresponding simulated time-domain Figures are shown in the middle row. The relaxation constants $T_2(ns)$ used for the various orientations are: (a) 800, (b) 812, (c) 687. A Gaussian inhomogeneous broadening width Δ=4 MHz, Eq. (4.34), is used in the simulation of the FT signal for each orientation.

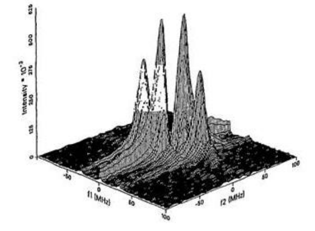
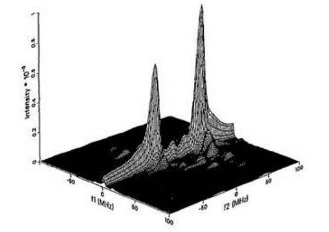
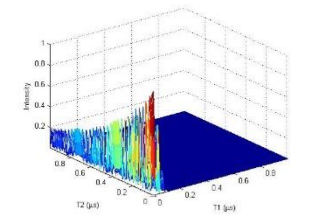
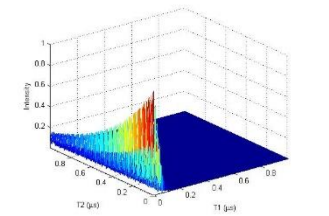
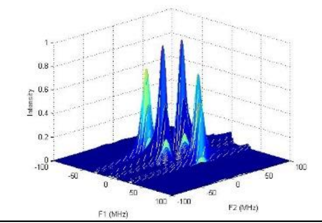
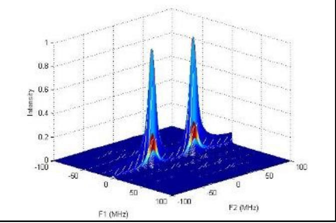
Experiment FT spectra		
Simulation Time-domain signal (this work)		
Simulation – Fourier transform (this work)		

Figure 4.6 Simulated time-domain and Fourier transforms of SECSY signals The experimental Figure reproduced by permission from Figure 9 of Lee *et al.* (1993) (upper row). Simulations (bottom) of SECSY spectra at (θ, ϕ) orientations of (a) (5°, 90°), (b) (45°, 90°) in the zy-quadrant $[(\alpha, \beta, \gamma) = (0, -\theta, 0)]$, in the second and third rows, respectively. The corresponding simulated time-domain Figures are shown in the middle row. The relaxation constants $T_2(ns)$ for the various orientations are (a) 1223, (b) 1068. A Gaussian inhomogeneous broadening width Δ=4 MHz, Eq.(4.34), is used in the simulation of FT signal for each orientation.

4.4.2 Echo-ELDOR

The ELDOR experiment measures the transfer of coherence from one EPR allowed transition to another. By including a mixing time in this three pulse sequence and transferring coherences to the z axis, this is sensitive to processes that occur on the T_1 time scale.

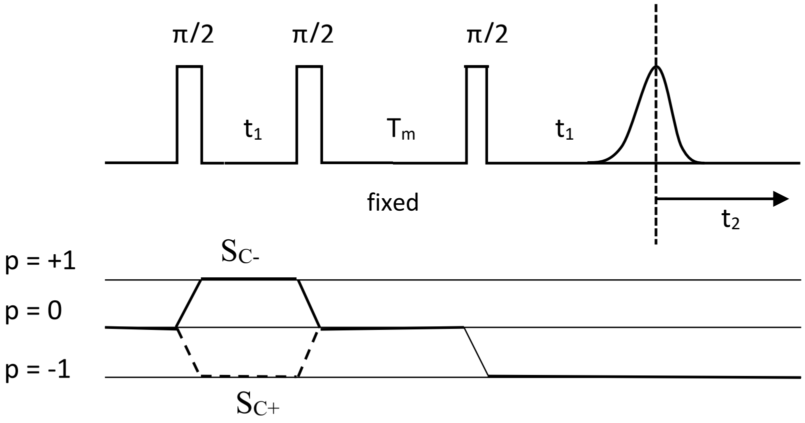
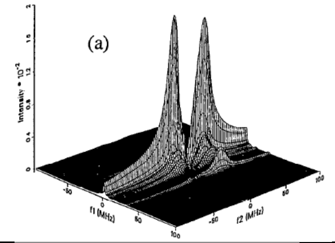
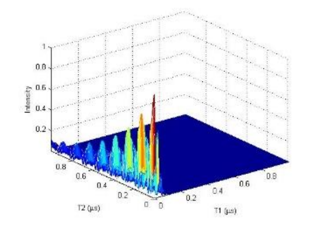
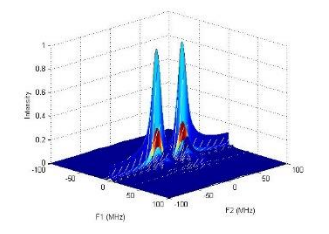
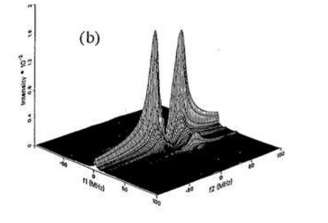
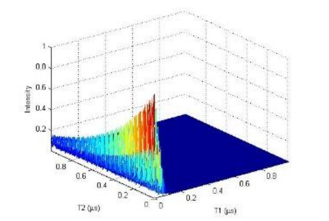
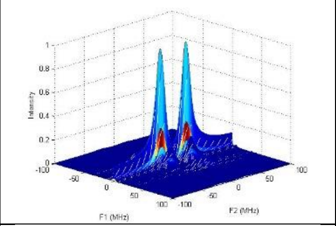
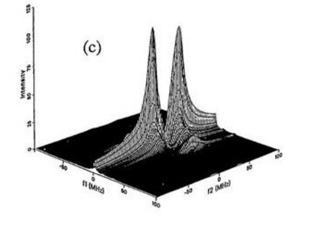
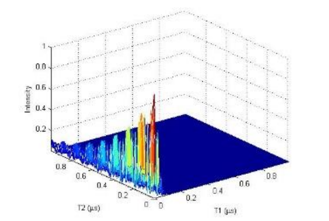
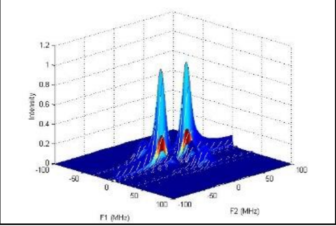


Figure 4.7 Pulse sequence for obtaining Echo- ELDOR signal

The t_1 time between the two pulses is stepped and this echo decay is measured as a function of t_2. Here T_m is the mixing time. (Bottom) Coherence pathways used for calculating echo-ELDOR signal for an unpaired electron (S = ½) interacting with a single nucleus (I = ½). p is the coherence order, which represents transverse magnetization, corresponding to spins rotating in a plane perpendicular to the external field (Lee *et al*., 1993).

As for the echo-ELDOR signals, they were calculated for the orientation $(\theta, \phi) = (30^{\circ}, 0^{\circ})$ in the zx-quadrant, so that$(\alpha, \beta, \gamma) = \ (0, -\theta, 0)$ (Lee *et al*., 1993), with four mixing times T_m (the fixed time interval between the second and third $\pi/2$ pulses; see Figure 4.8: (a) 5 μs; (b) 20 μs; (c) 40 μs; (d) 60 μs.

One can compare the experimental signals with the simulated signals. It is found that for the ELDOR signals simulated here, as shown in Figure 4.8, the positions of the main frequency peaks, i.e. the nuclear modulation frequencies $\omega_\alpha \approx 7.0MHz$ and $\omega_\beta \approx 32.0MHz$, are the same as those reported in Lee *et al*., 1993 in all four cases. As well, the shapes of the simulated spectra in the frequency domain as calculated here and those calculated by Lee *et al*., 1993 are found to be in good agreement with each other.

Experiment FT spectra	Simulation Time-domain signal (this work)	Simulation – Fourier transform (this work)
		
		
		

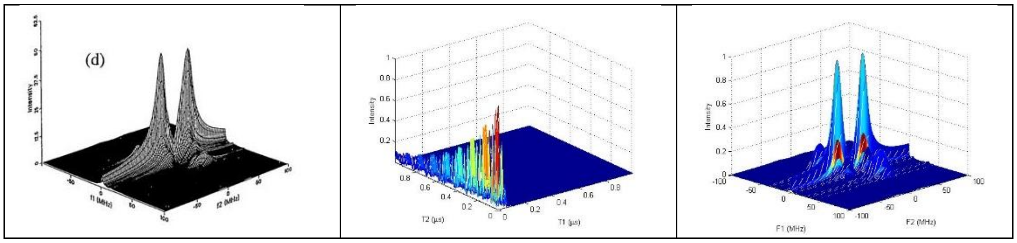


Figure 4.8 Simulated time-domain and Fourier transforms of Echo-ELDOR signals (Figure 11 of Lee *et al.* (1993*)* (reproduced here with the permission of the authors.)

The experimental Figures are reproduced by permission from Figure 11 of Lee *et al*. (1993) (first column). Simulations (second column) of the echo-ELDOR spectra at (θ, ϕ) orientations of (30°, 0°) in the zx-quadrant $[(\alpha, \beta, \gamma) = (0, -\theta, 0)]$, with the mixing times T_m: (a) 5 μs; (b) 20 μs; (c) 40 μs; (d) 60 μs. The corresponding FT Figures are shown in the last column. A Gaussian inhomogeneous broadening width Δ=4 MHz, Eq. (4.34), is used in the simulation.

4.5 Computing time

The computation work is undertaken on a laptop with an x64-based processor of Intel® Pentium® CPU B980 @2.4GHz, an installed memory (RAM) 4.00GB, an operation system of Windows 8.1, and a Matlab version 8.3.0.532 (R2014a). As for computational time, calculation of echo-ELDOR pulse signal takes ~10.56 seconds.

5. Simulation of Electron-electron dipolar interaction

5.1 Spin Hamiltonian and parameters

For the specific case of the electron-electron dipolar interaction of two nitroxide radicals, each with electron spin S=1/2, nuclear spin I=1/2, the composite Hamiltonian can be expressed as e.g. Misra (2011), Misra and Freed (2011):

$$\widehat{H}_0 = \widehat{H}_{01} + \widehat{H}_{02} + \widehat{H}_{12} \tag{5.1}$$

where

$$\widehat{H}_{0k} = S_{kz}\boldsymbol{g}_k \cdot \boldsymbol{B}_0 - \gamma_n I_{kz} B_0 + S_{kz}\boldsymbol{A}_k \cdot \boldsymbol{I}_k;\ k = 1, 2 \tag{5.2}$$

Ignoring the nuclear interaction, the various terms in Eq. (5.2) are:

$$\widehat{H}_{0k} = S_{kz}\boldsymbol{g}_k \cdot \boldsymbol{B}_0 \tag{5.3}$$

$$\widehat{H}_{12} = \widehat{H}_{D_{dip}} + \widehat{H}_J = \frac{D_{dip}}{2}(3cos^2\theta - 1)\left(S_z^2 - \frac{1}{3}\boldsymbol{S}^2\right) + J\left(\frac{1}{2} - \boldsymbol{S}_1 \cdot \boldsymbol{S}_2\right) \tag{5.4}$$

Here, in the direct-product space, $\boldsymbol{S} = \boldsymbol{S}_1 \otimes \boldsymbol{I}_2 + \boldsymbol{I}_1 \otimes \boldsymbol{S}_2$; $S_z = S_{z1} \otimes I_2 + I_1 \otimes S_{z2}$, J is the exchange-interaction constant between the two electrons, θ is the angle between the dipolar axis and the static magnetic field $\boldsymbol{B}_0$ and D_{dip} is the dipolar-coupling constant

$$D_{dip} = \frac{3\gamma_e^2\hbar}{2r^3} = \frac{3}{2}d \tag{5.5}$$

In the present case, neglecting the nuclear part, the direct-product Hilbert space is $S_1 \otimes S_2$, with the dimension $(2S_1 + 1) \times (2S_2 + 1) = 4$. Then the dimensions of the superoperator $\hat{\hat{\boldsymbol{H}}}_0$ in Liouville space are 16×16. [It is noted that if the nuclear space was taken into consideration the matrix dimension are $(16 \times 16) \times (16 \times 16) = 256 \times 256$, which would have been exorbitant for manipulation on a laptop.]

The results of calculation are shown in Figures. 5.1 and 5.2 for SECSY and echo-ELDOR cases. More details are given in the captions to these Figures.

5.2 Relaxation matrix

For consideration of relaxation for this illustrative case, only the relaxation of each electron with the lattice is taken into account. Otherwise, if the nuclear relaxations are taken into account, the Liouville space becomes of much larger dimension (256 x 256), which becomes unmanageable as explained above. One can add the nuclear relaxation to this phenomenologically in the same way as carried out in LPF (Lee *et al.*, 1993). As a consequence, for each electron the same relaxation matrix applies.

Then, the relaxation matrix for the two electrons, with the subscripts i=1, 2 designating the two electrons, is:

$$\left[\hat{\hat{\mathrm{R}}}_i\right] = \begin{bmatrix} \hat{\hat{\mathrm{R}}}_{aa,aa} & \hat{\hat{\mathrm{R}}}_{ab,ab} \\ \hat{\hat{\mathrm{R}}}_{ba,ba} & \hat{\hat{\mathrm{R}}}_{bb,bb} \end{bmatrix} = \begin{bmatrix} -W_e & \frac{1}{T_2} \\ \frac{1}{T_2} & -W_e \end{bmatrix} \tag{5.6}$$

The relaxation matrix for the electron-electron spin coupled dipolar system is then expressed as:

$$\hat{\hat{\mathrm{R}}} = \hat{\hat{\mathrm{R}}}_1 \otimes I_2 + I_1 \otimes \hat{\hat{\mathrm{R}}}_2 \tag{5.7}$$

In Eq.(5.7) I_1 and I_2 are 2x2 unit matrices for electrons 1 and 2, respectively.

5.3 Pulse-propagator superoperator for the dipolar-coupled system

For the two-electron dipolar-coupled system, the pulse propagator is the same as that expressed by (3.48).

In the $|S_z\rangle$ basis of a dipolar system with electron magnetic quantum numbers $M_1 = \pm 1/2\ and\ M_2 = \pm 1/2$, the explicit expression for $\hat{\varepsilon}(t)$ is as follows:

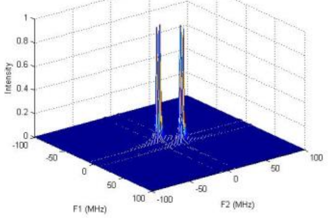
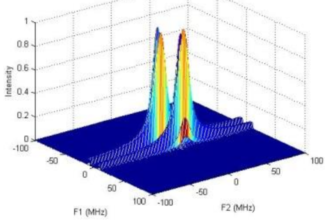
$$[\hat{\varepsilon}] = [\hat{\varepsilon}]_1 \otimes I_2 + I_1 \otimes [\hat{\varepsilon}]_2$$

$$
= \begin{array}{cc}
 & \begin{array}{cccc} |1\rangle = |+_1+_2\rangle & |2\rangle = |+_1-_2\rangle & |3\rangle = |-_1+_2\rangle & |4\rangle = |-_1-_2\rangle \end{array} \\
\begin{array}{c} |1\rangle = |+_1+_2\rangle \\ |2\rangle = |+_1-_2\rangle \\ |3\rangle = |-_1+_2\rangle \\ |4\rangle = |-_1-_2\rangle \end{array} & \frac{\omega}{2}\begin{bmatrix} 0 & e^{-i\phi} & e^{-i\phi} & 0 \\ e^{i\phi} & 0 & 0 & e^{-i\phi} \\ e^{i\phi} & 0 & 0 & e^{-i\phi} \\ 0 & e^{i\phi} & e^{i\phi} & 0 \end{bmatrix}
\end{array} \tag{5.8}
$$

5.4. Illustrative Examples

For two nitroxide radicals coupled by the dipolar interaction the time-domain SECSY and echo-ELDOR signals were calculated for the pulse sequences S_{c-}, neglecting the hyperfine and exchange interactions, otherwise the size of the matrix would have been prohibitively large in Liouville space [=(36x36)x(36x36)]. The value of the matrix $\tilde{g} = (g_{xx}, g_{yy}, g_{zz}) = (2.0026, 2.0035, 2.0033)$ was used for each electron as those used by Misra and Li (2018), and the value of the dipolar-interaction constant $D_{dip} = 10MHz$ was used (See Eq. (5.5) for the expression of D_{dip}) (Misra, 2011). Figure 5.1 shows the simulated Fourier transform of SECSY signals at (θ, ϕ) orientations of (30°, 0°) in the zx-quadrant, the corresponding values for the Euler angles (α, β, γ) are $(0, -\theta, 0)$. The lower row displays the FT of the slice of time domain along $t_2 = 0$; it clearly shows the dipolar peaks at $\pm 10MHz$, which corresponds, as expected, to the value of $\pm D_{dip} = \pm 10MHz$ used in the present simulation. Figure 5.2 shows the simulated Fourier transform of echo-ELDOR signals, as calculated for the mixing time $T_m = 20\mu s$; the lower row displays the Fourier transform of the echo peak (t_2=0 slice), which clearly shows the peaks at $\pm D_{dip} = \pm 10MHz$ as expected.

A Gaussian inhomogeneous broadening width Δ=4 MHz is used in the simulation. It is noted from Figures 5.1 and 5.2 that inclusion of relaxation has significantly broadened the dipolar peaks, as expected. However, the spacings of the Pake doublet in both cases, which are needed to estimate distances, remained unchanged. On the other hand, if constant-time experiments are to be done, then our calculation results without relaxation as shown in Figures 5.1 and 5.2 will be useful.

	Without Relaxation	With Relaxation
Fourier transform		

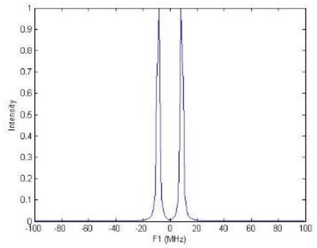
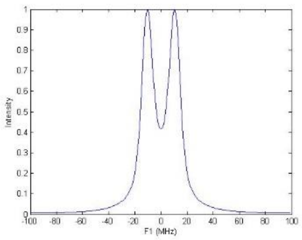
FT of the slice of time domain along t$_2$=0		

Figure 5.1 Simulated Fourier transform of SECSY signals with and without relaxation for an electron-electron dipolar-coupled system

Simulations at (θ,ϕ)orientations of (30°, 0°) in the zx-quadrant $[(\alpha,\beta,\gamma)=(0,-\theta,0)]$ (Lee *et al.*, 1993). The lower row is the Fourier transform of the echo peak (t$_2$=0 slice), which clearly shows the peaks at $\pm 10MHz$, the value of the dipolar coupling constant D_{dip} used in the simulation, as expected. It is noted that inclusion of relaxation significantly broadens the Fourier peaks, as expected. A Gaussian inhomogeneous broadening width Δ=4 MHz, is used in the simulation.

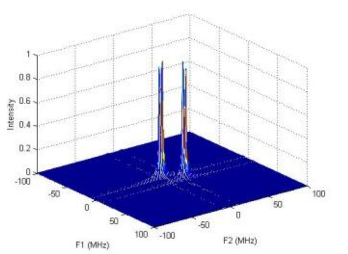
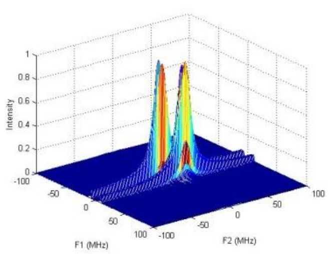
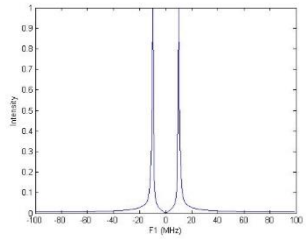
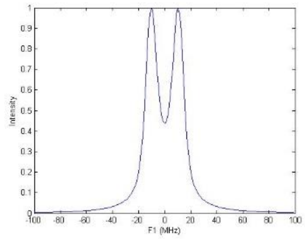
	Without Relaxation	With Relaxation
Fourier transform		
FT of the slice of time domain signal along t_2=0		

Figure 5.2 Fourier transform of simulated echo-ELDOR signals with and without relaxation for an electron-electron dipolar-coupled system.

Simulations at (θ, ϕ) orientations of (30°, 0°) in the zx-quadrant, so that $(\alpha, \beta, \gamma) = (0, -\theta, 0)$ (Lee *et al*., 1993), with the mixing times T_m = 20 μs. The lower row is the Fourier transform of the echo peak (t_2=0 slice), which clearly shows the dipolar peaks at $\pm 10MHz$, the value of the dipolar coupling constant D_{dip} used in the simulation. It is noted that inclusion of relaxation significantly broadens the Fourier peaks as expected. A Gaussian inhomogeneous broadening width Δ=4 MHz is used in the simulation.

6. Relaxation due to fluctuations of SHP

The time-dependent Hamiltonian $\widehat{H}_1(t)$ causing relaxation can be written in general form as follows (Abragam, 1961):

$$\widehat{H}_1(t) = \sum_{p,\alpha\beta} F^p_{\alpha\beta}(t) A^p_{\alpha\beta} \tag{6.1}$$

where the $A^p_{\alpha\beta}$ are the spin operators in the laboratory frame that appear in the spin Hamiltonian and their coefficients $F^p_{\alpha\beta}$ are functions of spatial variables, which are, in fact, the time-dependent fluctuating parts of SHPs, as given explicitly in Table IV of Lee *et al*. (1993). The various terms $F^p_{\alpha\beta}(t) = \left(\delta g, \delta a, \delta F, \delta F^{(2)}, \delta D, \delta D^{(2)}\right)$ in H_1 are the time-dependent fluctuations from the average values, e.g. $\delta g = g(t) - \bar{g},$; they are given in Appendix F below. These, in turn, bring about relaxation as discussed below.

6.1 Calculation of relaxation due to fluctuation of SHP

The matrix elements of $\hat{\hat{\mathrm{R}}}$ in can be extracted from the Redfield relaxation matrix (Redfield, 1957), as described by Freed and Fraenkel (1963). Let $\hat{\hat{\mathrm{R}}}_{\alpha\alpha'\beta\beta'}$ be the matrix elements of $\hat{\hat{\mathrm{R}}}$:

$$\begin{aligned}\hat{\hat{\hat{\mathrm{R}}}}_{\alpha\alpha'\beta\beta'} = {} & 2\, J_{\alpha\beta\alpha'\beta'}\left(\omega_{\alpha'\beta'}\right) - \delta_{\alpha'\beta'} \sum\nolimits_{\gamma} J_{\gamma\beta\gamma\alpha}\left(\omega_{\gamma\beta}\right) \\ & - \delta_{\alpha\beta} \sum\nolimits_{\gamma} J_{\gamma\alpha'\gamma\beta'}\left(\omega_{\gamma\beta'}\right)\end{aligned} \tag{6.2}$$

where the $J_{\alpha\beta\alpha'\beta'}(\omega)$ are the spectral density functions in the frequency domain, related to the autocorrelation function through the Fourier transformation:

$$J_{\alpha\beta\alpha'\beta'}(\omega) = \frac{1}{2}\int_{-\infty}^{\infty} e^{i\omega\tau} P_{\alpha\beta\alpha'\beta'}(\tau)\, d\tau \tag{6.3}$$

In Eq. (6.3) the autocorrelation matrix of H_1 is:

$$P_{\alpha\beta\alpha'\beta'}(\tau) = \langle H_{1\alpha\beta}(t) H^*_{1\alpha'\beta'}(t-\tau)\rangle \tag{6.4}$$

where $\langle\quad\rangle$ denotes the time average over the ensemble. It is noted that $P_{\alpha\beta\alpha'\beta'}(\tau)$ is independent of time, $P(\tau) = P(-\tau)$ and $P(\infty) = 0$ (Redfield, 1957) and the correlation time τ_c is governed by the condition that $P(\tau) \ll P(0)$ if $\tau \gg \tau_c$.

The following properties of $J_{\alpha\beta\alpha'\beta'}(\omega_{\alpha\beta})$ appearing in Eq.(6.2) (Freed and Fraenkel, 1963) are found to be useful:

$$J_{\alpha\beta\alpha'\beta'}(\omega_{\alpha\beta}) = J_{\alpha\beta\alpha'\beta'}(-\omega_{\alpha\beta}) = J_{\beta'\alpha'\beta\alpha}(\omega_{\alpha\beta}) = J^{*}_{\beta\alpha\beta'\alpha'}(\omega_{\alpha\beta}) \tag{6.5}$$

Using Eqs. (6.1) and (6.4), one obtains:

$$\begin{aligned} P_{\alpha\beta\alpha'\beta'}(\tau) &= \sum_{p,q\alpha\beta,\alpha'\beta'} \langle F^{p}_{\alpha\beta}(t)A^{p}_{\alpha\beta}\left(F^{q}_{\alpha\beta}(t-\tau)A^{q}_{\alpha\beta}(t-\tau)\right)^{*}\rangle \\ &= \sum_{p,q,\alpha\beta,\alpha'\beta'} A^{p}_{\alpha\beta}\left(A^{q}_{\alpha'\beta'}\right)^{*}\langle F^{q}_{\alpha\beta}(t)\left(F^{q}_{\alpha\beta}(t-\tau)\right)^{*}\rangle \end{aligned} \tag{6.6}$$

On the other hand, eqs. (6.3) and (6.6) yield

$$\begin{aligned} J_{\alpha\beta\alpha'\beta'}(\omega) = \frac{1}{2}\sum_{p,q} A^{p}_{\alpha\beta}\left(A^{q}_{\alpha'\beta'}\right)^{*}\int_{-\infty}^{\infty} e^{i\omega\tau}\,\langle F^{p}_{\alpha\beta}(t)\Big(F^{q}_{\alpha\beta}(t \\ - \tau)\Big)^{*}\rangle\, d\tau \end{aligned} \tag{6.7}$$

$$J_{\alpha\alpha\beta\beta}(0) = \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\alpha\alpha}(t)\left(A^{q}_{\beta\beta}(t)\right)^{*} F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau \tag{6.8}$$

$$J_{\alpha\alpha\alpha\alpha}(0) = \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\alpha\alpha}(t)\left(A^{q}_{\alpha\alpha}(t-\tau)\right)^{*} F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau \tag{6.9}$$

$$J_{\beta\beta\beta\beta}(0) = \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\beta\beta}(t)\left(A^{q}_{\beta\beta}(t-\tau)\right)^{*} F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau \tag{6.10}$$

In the $\widehat{H}_0$ basis with a significant magnitude of hf tensor, there result no degenerate energy levels; as well, there occur no transitions between any of these energy levels, so only the following specific values for the matrix elements in Eq. (6.2) are required.

6.1.1 Calculation of element of relaxation matrix

We illustrate how to calculate an element of relaxation matrix

$$\begin{aligned} \hat{\hat{R}}_{\alpha\beta\alpha\beta} = {} & 2\,J_{\alpha\alpha\beta\beta}(\omega_{\beta\beta}) \;-\delta_{\beta\beta}\sum\nolimits_{\gamma} J_{\gamma\alpha\gamma\alpha}\,(\omega_{\gamma\alpha}) - \delta_{\alpha\alpha}\sum\nolimits_{\gamma} J_{\gamma\beta\gamma\beta}(\omega_{\gamma\beta}) \\ & = \;2\,J_{\alpha\alpha\beta\beta}(0) - J_{\alpha\alpha\alpha\alpha}(0)\; - J_{\beta\beta\beta\beta}(0) - \sum\nolimits_{\gamma\neq\alpha} J_{\gamma\alpha\gamma\alpha}\,(\omega_{\gamma\alpha}) \\ & -\sum\nolimits_{\gamma\neq\alpha} J_{\gamma\beta\gamma\beta}(\omega_{\gamma\beta}) \end{aligned} \tag{6.11}$$

We separate the calculation:

$$\begin{aligned}
&2\,J_{\alpha\alpha\beta\beta}(0)-J_{\alpha\alpha\alpha\alpha}(0)\ -J_{\beta\beta\beta\beta}(0)\\
&\qquad = J_{\alpha\alpha\beta\beta}(0)-J_{\alpha\alpha\alpha\alpha}(0)+J_{\alpha\alpha\beta\beta}(0)-J_{\beta\beta\beta\beta}(0)\\
&\qquad =\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A_{\alpha\alpha}^{p}(t)\left(A_{\beta\beta}^{q}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\,d\tau\\
&\qquad -\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A_{\alpha\alpha}^{p}(t)\left(A_{\alpha\alpha}^{q}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\,d\tau\\
&\qquad +\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A_{\alpha\alpha}^{p}(t)\left(A_{\beta\beta}^{q}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\,d\tau\\
&\qquad -\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A_{\beta\beta}^{p}(t)\left(A_{\beta\beta}^{q}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\,d\tau
\end{aligned}\tag{6.12}$$

$$\begin{aligned}
&=\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A_{\alpha\alpha}^{p}(t)\left(A_{\beta\beta}^{q}(t-\tau)-A_{\alpha\alpha}^{q}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\,d\tau\\
&\qquad -\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A_{\beta\beta}^{p}(t)\right.\\
&\qquad \left.-A_{\alpha\alpha}^{p}(t)\right)\left(A_{\beta\beta}^{q}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\,d\tau
\end{aligned}\tag{6.13}$$

The autocorrelation function

$$P_{\alpha\beta\alpha'\beta'}(\tau)=\langle H_{1\alpha\beta}(t)H_{1\alpha'\beta'}^{*}(t-\tau)\rangle\tag{6.14}$$

has the symmetry property (Redfield, 1957)

$$P_{\alpha\beta\alpha'\beta'}(\tau)=P_{\alpha\beta\alpha'\beta'}(-\tau)\tag{6.15}$$

thus

$$\begin{aligned}
&=\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A_{\beta\beta}^{p}(t)-A_{\alpha\alpha}^{p}(t)\right)\left(A_{\beta\beta}^{q}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\,d\tau\\
&\qquad =\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A_{\beta\beta}^{p}(t)\right.\\
&\qquad \left.-A_{\alpha\alpha}^{p}(t)\right)\left(A_{\beta\beta}^{q}(t+\tau)\right)^{*}F^{p}(t)\left(F^{q}(t+\tau)\right)^{*}\rangle\,d\tau
\end{aligned}\tag{6.16}$$

Changing t to $t-\tau$, one get

$$=\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A_{\beta\beta}^{p}(t-\tau)-A_{\alpha\alpha}^{p}(t-\tau)\right)\left(A_{\beta\beta}^{q}(t)\right)^{*}F^{p}(t-\tau)\left(F^{q}(t)\right)^{*}\rangle\,d\tau\tag{6.17}$$

rename p, q

$$\begin{aligned}&= \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A^{p}_{\beta\beta}(t)\right)^{*}\left(A^{q}_{\beta\beta}(t-\tau)-A^{q}_{\alpha\alpha}(t-\tau)\right)\left(F^{p}(t)\right)^{*}F^{q}(t-\tau)\rangle\, d\tau = \\ &= \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\Big[A^{p}_{\beta\beta}(t)\left(A^{q}_{\beta\beta}(t-\tau) \right. \\ &\qquad \left. - A^{q}_{\alpha\alpha}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\Big]^{*}\rangle\, d\tau\end{aligned} \tag{6.18}$$

and with fact that autocorrelation function are real

$$\begin{aligned}&\frac{1}{2}\left[\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\beta\beta}(t)\left(A^{q}_{\beta\beta}(t-\tau)-A^{q}_{\alpha\alpha}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau\right]^{*} \\ &\qquad = \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\beta\beta}(t)\left(A^{q}_{\beta\beta}(t-\tau) \right. \\ &\qquad \left. - A^{q}_{\alpha\alpha}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau\end{aligned} \tag{6.19}$$

thus

$$\begin{aligned}&\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\alpha\alpha}(t)\left(A^{q}_{\beta\beta}(t-\tau)-A^{q}_{\alpha\alpha}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau \\ &\qquad -\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A^{p}_{\beta\beta}(t) \right. \\ &\qquad \left. - A^{p}_{\alpha\alpha}(t)\right)\left(A^{q}_{\beta\beta}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau \\ &\qquad = \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\alpha\alpha}(t)\left(A^{q}_{\beta\beta}(t-\tau) \right. \\ &\qquad \left. - A^{q}_{\alpha\alpha}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau \\ &\qquad -\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle A^{p}_{\beta\beta}(t)\left(A^{q}_{\beta\beta}(t-\tau) \right. \\ &\qquad \left. - A^{q}_{\alpha\alpha}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau\end{aligned} \tag{6.20}$$

$$\begin{aligned}&= \frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A^{p}_{\alpha\alpha}(t) \right. \\ &\qquad \left. - A^{p}_{\beta\beta}(t)\right)\left(A^{q}_{\beta\beta}(t-\tau)-A^{q}_{\alpha\alpha}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau \\ &\qquad = -\frac{1}{2}\sum_{p,q}\int_{-\infty}^{\infty}\langle\left(A^{p}_{\alpha\alpha}(t) \right. \\ &\qquad \left. - A^{p}_{\beta\beta}(t)\right)\left(A^{q}_{\alpha\alpha}(t-\tau)-A^{q}_{\beta\beta}(t-\tau)\right)^{*}F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle\, d\tau\end{aligned} \tag{6.21}$$

$$
\begin{aligned}
&= -\frac{1}{2}\int_{-\infty}^{\infty} \langle \sum\nolimits_p \Big[A_{\alpha\alpha}^p(t) \\
&\qquad - A_{\beta\beta}^p(t)F^p(t)\Big] \sum\nolimits_q \Big[A_{\alpha\alpha}^q(t-\tau) - A_{\beta\beta}^q(t-\tau)\big(F^q(t-\tau)\big)\Big]^* \rangle\, d\tau \\
&\qquad = -\int_0^{\infty} \langle \sum\nolimits_p \Big[A_{\alpha\alpha}^p(t) \\
&\qquad - A_{\beta\beta}^p(t)F^p(t)\Big] \sum\nolimits_q \Big[A_{\alpha\alpha}^q(t-\tau) - A_{\beta\beta}^q(t-\tau)\big(F^q(t-\tau)\big)\Big]^* \rangle\, d\tau
\end{aligned}
\tag{6.22}
$$

with

$$
\widehat{H}_1(t) = \sum_p F^p(t) A^p \tag{6.23}
$$

one gets

$$
\sum\nolimits_p \Big[A_{\alpha\alpha}^p(t) - A_{\beta\beta}^p(t)F^p(t)\Big] = \big[\widehat{H}_{1\alpha\alpha}(t) - \widehat{H}_{1\beta\beta}(t)\big] = \omega_{\alpha\beta}(t) \tag{6.24}
$$

$$
\begin{aligned}
\hat{\hat{R}}_{\alpha\beta\alpha\beta} = &-\sum_{p,q}\int_0^{\infty} \langle \Big(A_{\alpha\alpha}^p(t) \\
&- A_{\beta\beta}^p(t)\Big) F^p(t) \Big(A_{\alpha\alpha}^q(t-\tau) - A_{\beta\beta}^q(t-\tau)\Big)^* \big(F^q(t-\tau)\big)^* \rangle\, d\tau \\
&- \sum\nolimits_{\gamma\neq\alpha} J_{\gamma\alpha\gamma\alpha}\,(\omega_{\gamma\alpha}) - \sum\nolimits_{\gamma\neq\alpha} J_{\gamma\beta\gamma\beta}(\omega_{\gamma\beta}) \\
&= -\int_0^{\infty} \langle \omega_{\alpha\beta}(t)\omega_{\alpha\beta}^*(t-\tau)\rangle d\tau - \sum\nolimits_{\gamma\neq\alpha} J_{\gamma\alpha\gamma\alpha}\,(\omega_{\gamma\alpha}) \\
&- \sum\nolimits_{\gamma\neq\alpha} J_{\gamma\beta\gamma\beta}(\omega_{\gamma\beta})
\end{aligned}
\tag{6.25}
$$

because $A_{\alpha\alpha}^p, A_{\beta\beta}^p, A_{\alpha\alpha}^q, A_{\beta\beta}^q$ are real and time independent:

$$
\begin{aligned}
\hat{\hat{R}}_{\alpha\beta\alpha\beta} = &-\frac{1}{2}\sum_{p,q}\Big(A_{\alpha\alpha}^p - A_{\beta\beta}^p\Big)\Big(A_{\alpha\alpha}^q - A_{\beta\beta}^q\Big)\int_{-\infty}^{\infty} \langle F^p(t)\big(F^q(t-\tau)\big)^* \rangle d\tau \\
&- \sum\nolimits_{\gamma\neq\alpha} J_{\gamma\alpha\gamma\alpha}\,(\omega_{\gamma\alpha}) - \sum\nolimits_{\gamma\neq\alpha} J_{\gamma\beta\gamma\beta}(\omega_{\gamma\beta})
\end{aligned}
\tag{6.26}
$$

$$
\begin{aligned}
\hat{\hat{\mathrm{R}}}_{\alpha\alpha\beta\beta} &= J_{\alpha\beta\alpha\beta}(\omega_{\alpha\beta}) + J_{\alpha\beta\alpha\beta}(\omega_{\alpha\beta}) - \delta_{\alpha\beta}\sum\nolimits_{\gamma} J_{\gamma\beta\gamma\alpha}\,(\omega_{\gamma\beta}) \\
&\quad - \delta_{\alpha\beta}\sum\nolimits_{\gamma} J_{\gamma\alpha\gamma\beta}(\omega_{\gamma\beta})
\end{aligned}
\tag{6.27}
$$

It is seen from Eq.(6.27) that if $\alpha \neq \beta$,

$$
\hat{\hat{\mathrm{R}}}_{\alpha\alpha\beta\beta} = 2J_{\alpha\beta\alpha\beta}(\omega_{\alpha\beta}) \tag{6.28}
$$

and

$$\begin{aligned}\hat{\hat{R}}_{\alpha\alpha\beta\beta} &= 2J_{\alpha\beta\alpha\beta}(\omega_{\alpha\beta}) \\ &= \sum_{p,q}\int_{-\infty}^{\infty} A_{\alpha\beta}^{p}(t)\left(A_{\alpha\beta}^{q}(t-\tau)\right)^{*} e^{i\omega_{\alpha\beta}\tau}\langle F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle d\tau \\ &= \sum_{p,q} A_{\alpha\beta}^{p}A_{\alpha\beta}^{q}\int_{-\infty}^{\infty} e^{i\omega_{\alpha\beta}\tau}\langle F^{p}(t)\left(F^{q}(t-\tau)\right)^{*}\rangle d\tau\end{aligned} \tag{6.29}$$

and from Eq. (6.5), one obtains

$$\hat{\hat{\mathrm{R}}}_{\alpha\alpha\beta\beta} = \hat{\hat{\mathrm{R}}}_{\beta\beta\alpha\alpha} \tag{6.30}$$

Equations (6.27) - (6.30) yield

$$\hat{\hat{\mathrm{R}}}_{\alpha\alpha\alpha\alpha} = -\sum_{\beta(\neq\alpha)} \hat{\hat{\mathrm{R}}}_{\alpha\alpha\beta\beta} \tag{6.31}$$

From Eq. (6.2) by relabeling α', β, β' as β, α, β, respectively, one obtains

$$\begin{aligned}\hat{\hat{\mathrm{R}}}_{\alpha\beta\alpha\beta} = 2J_{\alpha\alpha\beta\beta}(\omega_{\beta\beta} = 0) - J_{\alpha\alpha\alpha\alpha}(0) - J_{\beta\beta\beta\beta}(0) \\ + \frac{1}{2}\left(\hat{\hat{\mathrm{R}}}_{\alpha\alpha\alpha\alpha} + \hat{\hat{\mathrm{R}}}_{\beta\beta\beta\beta}\right)\end{aligned} \tag{6.32}$$

Now using Eq. (6.5) after relabeling indices $\beta', \alpha', \beta,$ as β, β, α, respectively, one obtains

$$J_{\alpha\alpha\beta\beta}(0) = J_{\beta\beta\alpha\alpha}(0) \tag{6.33}$$

so that from Eqs. (6.32) and (6.33), it is seen that

$$\hat{\hat{\mathrm{R}}}_{\alpha\beta\alpha\beta} = \hat{\hat{\mathrm{R}}}_{\beta\alpha\beta\alpha} \tag{6.34}$$

Except for the above, all other elements

$$\hat{\hat{R}}_{\alpha\beta\alpha'\beta'}(\omega_{\eta\xi}) = 0 \tag{6.35}$$

Finally, the above evaluations imply that one only has to compute the spectral density matrix elements $J_{\alpha\beta\alpha\beta}(\omega_{\alpha\beta})$, $J_{\alpha\alpha\alpha\alpha}(0)$, and $J_{\alpha\alpha\beta\beta}(0)$ to consider the effect of fluctuations on relaxation.

6.2 Autocorrelation and spectral density functions

Calculation of ensemble average of elements of autocorrelation matrix, $P_{\alpha\alpha'\beta\beta'}$, and its spectral density function, $J_{\alpha\alpha'\beta\beta'}$, due to the fluctuating perturbation are discussed by Slichter (2013) , Eq. (5.294), which states

$$\langle F_{\alpha\beta}^{p}(t)\left(F_{\alpha\beta}^{p}(t-\tau)\right)^{*}\rangle = f_{\alpha\beta}^{p}e^{-|\tau|/\tau_c} \tag{6.36}$$

where $f_{\alpha\beta}^{p}$ is the amplitude of the function $F_{\alpha\beta}^{p}(t)$.

Noting that if $p \neq q$

$$\langle F_{\alpha\beta}^{p}(t)\left(F_{\alpha\beta}^{q}(t-\tau)\right)^{*}\rangle = 0 \tag{6.37}$$

as shown by Slitcher (2013) (Eq. (5.344)).

From Eqs.(6.7) and (6.36), one obtains

$$\begin{aligned} J_{\alpha\beta\alpha'\beta'}(\omega) &= \frac{1}{2}\sum_{p} A^{p}_{\alpha\beta}\left(A^{p}_{\alpha'\beta'}\right)^{*}\int_{-\infty}^{\infty} e^{i\omega\tau} f_p^2 e^{-|\tau|/\tau_c} d\tau \\ &= \frac{\tau_c}{1+(\omega\tau_c)^2}\sum_{p=1}^{n} A^{p}_{\alpha\beta}\left(A^{p}_{\alpha'\beta'}\right)^{*} f_p^2 \end{aligned} \tag{6.38}$$

From Eqs. (6.7) and (6.36), the non-zero elements of the spectral density function are calculated to be as follows:

$$\begin{aligned} J_{\alpha\beta\alpha\beta}\left(\omega_{\alpha\beta}\right) &= \frac{\tau_c}{1+\left(\omega_{\alpha\beta}\tau_c\right)^2}\sum_{q=1}^{n}\left|A^{p}_{\alpha\beta}\right|^2 f_p^2 \\ J_{\alpha\alpha\beta\beta}(0) &= \tau_c\sum_{p=1}^{n} A^{p}_{\alpha\alpha}\left(A^{p}_{\beta\beta}\right)^{*} f_p^2 \\ J_{\alpha\alpha\alpha\alpha}(0) &= \tau_c\sum_{q=1}^{n}\left|A^{p}_{\alpha\alpha}\right|^2 f_p^2 \end{aligned} \tag{6.39}$$

In Eqs. (6.38) and (6.39) each $\alpha, \beta, \alpha', \beta'$ corresponding to the eigenvalues of H_0 as described in Appendix A.

6.3 Relaxation due to fluctuation of SHP for an electron-nuclear spin-coupled system

The effect of the relaxation due to the time-dependent fluctuating perturbation is here taken into account quantitatively by using the procedure outlined by Lee *et al*. (1993). The specific role of the g and hf tensors is itemized in Table IV of Lee *et al*. (1993) in the $\widehat{H}_0$ basis. Note that not only are the spin operators F^p orientation dependent, but their coefficients $A^{p}_{\alpha\beta}$ can also have substantial orientation dependence.

The relaxation superoperator matrix $\hat{\hat{\mathrm{R}}}$ will, in general, display orientation dependence. In problems which involve orientational motion (Fresch *et al*., 2006; Libertini and Griffith, 1970), $\hat{\hat{\mathrm{R}}}$ becomes a rotational diffusion operator and it does not commute with the orientation dependent terms in $\widehat{H}$.

For the present calculation in a malonic acid crystal, it is assumed that phonon modulation of the hf and g tensors leads to spin relaxation that can be described by the Redfield equation applicable to motional narrowing. Then Eq. (6.1) for $\widehat{H}_1$ is the thermal average over these fluctuations, and the Redfield relaxation matrix $\hat{\hat{\mathrm{R}}}$ is composed of the small fluctuations in these terms, which are summarized in Table IV of Lee *et al*. (1993). It is noted that that in this approach, the density matrix for each orientation of the molecular magnetic tensors relative to the lab frame specified by $\Omega = (0, \beta, \gamma)$ is taken as uncoupled from all others.

This simplification would not be strictly correct if, for example, there were torsional oscillations, the Redfield relaxation matrix will then depend on the orientation of the single crystal. Thus a separate value of $\hat{\hat{\mathrm{R}}}$ is considered for each value of Ω in the corresponding $\widehat{H}_0$ basis. (This is a local approach as opposed to the global approach utilized in motional narrowing problems in which $\hat{\hat{\mathrm{R}}}$ is expressed in the $|S_z I_z\rangle$ basis.). Finally, the effects of spin relaxation is taken into account phenomenologically by the Redfield equation (Redfield, 1957) for an arbitrary orientation Ω.

6.3.1 Relaxation matrix $\hat{\hat{\mathrm{R}}}$

$$\hat{\hat{R}}_{acac} = 2J_{aacc}(\omega_{cc}) - \delta_{cc}\sum_{\gamma} J_{\gamma a\gamma a}(\omega_{\gamma a}) - \delta_{aa}\sum_{\gamma} J_{\gamma c\gamma c}(\omega_{\gamma b}) \tag{6.40}$$

and

$$\begin{aligned}\hat{\hat{R}}_{acac} = {} & 2\tau_c\sum_{p=1}^{n} A_{aa}^{p}\left(A_{cc}^{p}\right)^{*} f_p^2 - \tau_c\sum_{p=1}^{n}\left|A_{aa}^{p}\right|^2 f_p^2 - \tau_c\sum_{p=1}^{n}\left|A_{cc}^{p}\right|^2 f_p^2 \\ & - \frac{\tau_c}{1+(\omega_{ba}\tau_c)^2}\sum_{p=1}^{n}\left|A_{ba}^{p}\right|^2 f_p^2 - \frac{\tau_c}{1+(\omega_{ca}\tau_c)^2}\sum_{p=1}^{n}\left|A_{ca}^{p}\right|^2 f_p^2 \\ & - \frac{\tau_c}{1+(\omega_{da}\tau_c)^2}\sum_{p=1}^{n}\left|A_{da}^{p}\right|^2 f_p^2 - \frac{\tau_c}{1+(\omega_{ac}\tau_c)^2}\sum_{p=1}^{n}\left|A_{ac}^{p}\right|^2 f_p^2 \\ & - \frac{\tau_c}{1+(\omega_{bc}\tau_c)^2}\sum_{p=1}^{n}\left|A_{bc}^{p}\right|^2 f_p^2 - \frac{\tau_c}{1+(\omega_{dc}\tau_c)^2}\sum_{p=1}^{n}\left|A_{dc}^{p}\right|^2 f_p^2\end{aligned} \tag{6.41}$$

If all Spin-Hamiltonian Parameters (SHP) have same average fluctuation rate $\Delta(SHP)$, then

$$f_p = \Delta(SHP) f_{0p} \tag{6.42}$$

$$\begin{aligned}\hat{\hat{R}}_{acac} = \tau_c\left(\Delta(SHP)\right)^2 \Bigg[& 2\sum_{p=1}^{n} A_{aa}^{p}\left(A_{cc}^{p}\right)^{*} f_{0p}^2 - \sum_{p=1}^{n}\left|A_{aa}^{p}\right|^2 f_{0p}^2 - \sum_{p=1}^{n}\left|A_{cc}^{p}\right|^2 f_{0p}^2 \\ & - \frac{1}{1+(\omega_{ba}\tau_c)^2}\sum_{p=1}^{n}\left|A_{ba}^{p}\right|^2 f_{0p}^2 - \frac{1}{1+(\omega_{ca}\tau_c)^2}\sum_{p=1}^{n}\left|A_{ca}^{p}\right|^2 f_{0p}^2 \\ & - \frac{1}{1+(\omega_{da}\tau_c)^2}\sum_{p=1}^{n}\left|A_{da}^{p}\right|^2 f_{0p}^2 - \frac{1}{1+(\omega_{ac}\tau_c)^2}\sum_{p=1}^{n}\left|A_{ac}^{p}\right|^2 f_{0p}^2 \\ & - \frac{1}{1+(\omega_{bc}\tau_c)^2}\sum_{p=1}^{n}\left|A_{bc}^{p}\right|^2 f_{0p}^2 - \frac{1}{1+(\omega_{dc}\tau_c)^2}\sum_{p=1}^{n}\left|A_{dc}^{p}\right|^2 f_{0p}^2 \Bigg]\end{aligned} \tag{6.43}$$

then

$$\hat{\hat{R}}_{acac} \propto \tau_c\left(\Delta(SHP)\right)^2 \tag{6.44}$$

with

$$\hat{\hat{R}}_{abab} = \; 2\, J_{aabb}(\omega_{bb}) \;\; - \delta_{bb} \sum\nolimits_{\gamma} J_{\gamma a \gamma a}\left(\omega_{\gamma a}\right) - \delta_{aa} \sum\nolimits_{\gamma} J_{\gamma b \gamma b}\left(\omega_{\gamma b}\right) \quad (6.45)$$

and

$$\begin{aligned}\hat{\hat{R}}_{abab} = 2\tau_c \sum_{p=1}^{n} A_{aa}^{p}\left(A_{bb}^{p}\right)^{*} f_p^2 - \tau_c \sum_{p=1}^{n} \left|A_{\alpha\alpha}^{p}\right|^2 f_p^2 - \tau_c \sum_{p=1}^{n} \left|A_{bb}^{p}\right|^2 f_p^2 \\ - \frac{2\tau_c}{1+(\omega_{ba}\tau_c)^2} \sum_{p=1}^{n} \left|A_{ba}^{p}\right|^2 f_p^2 - \frac{\tau_c}{1+(\omega_{ca}\tau_c)^2} \sum_{p=1}^{n} \left|A_{ca}^{p}\right|^2 f_p^2 \\ - \frac{\tau_c}{1+(\omega_{da}\tau_c)^2} \sum_{p=1}^{n} \left|A_{da}^{p}\right|^2 f_p^2 - \frac{\tau_c}{1+(\omega_{cb}\tau_c)^2} \sum_{p=1}^{n} \left|A_{cb}^{p}\right|^2 f_p^2 \\ - \frac{\tau_c}{1+(\omega_{db}\tau_c)^2} \sum_{p=1}^{n} \left|A_{db}^{p}\right|^2 f_p^2\end{aligned} \quad (6.46)$$

If all Spin-Hamiltonian Parameters (SHP) have same average fluctuation rate $\Delta(SHP)$, then

$$\begin{aligned}\hat{\hat{R}}_{abab} = \tau_c\big(\Delta(SHP)\big)^2 \Bigg[2 \sum_{p=1}^{n} A_{aa}^{p}\left(A_{bb}^{p}\right)^{*} f_{0p}^2 - \sum_{p=1}^{n} \left|A_{\alpha\alpha}^{p}\right|^2 f_{0p}^2 - \sum_{p=1}^{n} \left|A_{bb}^{p}\right|^2 f_{0p}^2 \\ - \frac{2}{1+(\omega_{ab}\tau_c)^2} \sum_{p=1}^{n} \left|A_{ab}^{p}\right|^2 f_{0p}^2 - \frac{1}{1+(\omega_{ca}\tau_c)^2} \sum_{p=1}^{n} \left|A_{ca}^{p}\right|^2 f_{0p}^2 \\ - \frac{1}{1+(\omega_{da}\tau_c)^2} \sum_{p=1}^{n} \left|A_{da}^{p}\right|^2 f_{0p}^2 - \frac{1}{1+(\omega_{cb}\tau_c)^2} \sum_{p=1}^{n} \left|A_{cb}^{p}\right|^2 f_{0p}^2 \\ - \frac{1}{1+(\omega_{db}\tau_c)^2} \sum_{p=1}^{n} \left|A_{db}^{p}\right|^2 f_{0p}^2 \Bigg]\end{aligned} \quad (6.47)$$

and

$$\hat{\hat{R}}_{abab} \propto \tau_c\big(\Delta(SHP)\big)^2 \quad (6.48)$$

6.3.2 Spectral density functions

Using Eq. (6.39) and table IV of Lee *et al*. (1993) as given here in Appendix F, one can express the various $J_{\alpha\beta\alpha'\beta'}$ explicitly as follows:

$$
\begin{aligned}
J_{aabb}(\omega_{ab}) = \frac{\tau_c \gamma_e}{4\hbar} & \\
& \times \frac{1}{1+(\omega_{ab}\tau_c)^2}\Bigg\{a^2[Re(c_1^* c_2)]^2 \\
& + \Bigg[\delta D\,\frac{1}{2}(3\cos^2\beta - 1) \\
& + \delta D^{(2)}\sin^2\beta\cos(2\gamma)\Bigg]^2\,[Re(c_1^* c_2)]^2 \\
& + \Bigg[\delta D\,\frac{3}{8}\sin(2\beta) - \delta D^{(2)}\,\frac{1}{4}\sin(2\beta)\cos(2\gamma)\Bigg]^2\,(|c_1|^2 \\
& - |c_2|^2)^2\Bigg\}
\end{aligned}
\tag{6.49}
$$

$$
\begin{aligned}
J_{aacc}(\omega_{ac}) = \frac{\tau_c}{1+(\omega_{ac}\tau_c)^2}\Bigg\{&\Bigg[\delta F\,\frac{3}{8}\sin(2\beta) - \delta F^{(2)}\,\frac{1}{4}\sin(2\beta)\cos(2\gamma)\Bigg]^2\,m_1^2 \\
& + \left(-\frac{\gamma_e}{\hbar}a\right)^2\frac{1}{4}|c_2^* c_3|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D\,\frac{1}{2}(3\cos^2\beta - 1) \\
& + \delta D^{(2)}\sin^2\beta\cos(2\gamma)\Bigg]\Bigg]^2\,\frac{1}{16}|c_2^* c_3|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D\,\frac{3}{8}\sin(2\beta) - \delta D^{(2)}\,\frac{1}{4}\sin(2\beta)\cos(2\gamma)\Bigg]\Bigg]^2\,\frac{1}{4}|c_1^* c_3 \\
& - c_2^* c_4|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D\,\frac{3}{8}\sin^2 \\
& - \delta D^{(2)}\,\frac{1}{4}(1+\cos^2\beta)\cos(2\gamma)\Bigg]\Bigg]^2\,|c_1^* c_4|^2\Bigg\}
\end{aligned}
\tag{6.50}
$$

$$
\begin{aligned}
J_{bbdd}(\omega_{bd}) = \frac{\tau_c}{1+(\omega_{bd}\tau_c)^2}\Bigg\{&\Bigg[\delta F\,\frac{3}{8}\sin(2\beta) - \delta F^{(2)}\,\frac{1}{4}\sin(2\beta)\cos(2\gamma)\Bigg]^2\,m_1^2 \\
& + \left(-\frac{\gamma_e}{\hbar}a\right)^2\frac{1}{4}|c_1^* c_4|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D\,\frac{1}{2}(3\cos^2\beta - 1) \\
& + \delta D^{(2)}\sin^2\beta\cos(2\gamma)\Bigg]\Bigg]^2\,\frac{1}{16}|c_1^* c_4|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D\,\frac{3}{8}\sin(2\beta) - \delta D^{(2)}\,\frac{1}{4}\sin(2\beta)\cos(2\gamma)\Bigg]\Bigg]^2\,\frac{1}{4}|c_1^* c_3 \\
& - c_2^* c_4|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D\,\frac{3}{8}\sin^2 \\
& - \delta D^{(2)}\,\frac{1}{4}(1+\cos^2\beta)\cos(2\gamma)\Bigg]\Bigg]^2\,|c_2^* c_3|^2\Bigg\}
\end{aligned}
\tag{6.51}
$$

$$
\begin{aligned}
J_{aadd} = \frac{\tau_c}{1+(\omega_{ad}\tau_c)^2} \Bigg\{ & \left[\delta F \frac{3}{8}\sin(2\beta) - \delta F^{(2)} \frac{1}{4}\sin(2\beta)\cos(2\gamma)\right]^2 m_2^2 \\
& + \left(-\frac{\gamma_e}{\hbar} a\right)^2 \frac{1}{4} |c_2^* c_4|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D \frac{1}{2}(3\cos^2\beta - 1) \\
& + \delta D^{(2)} \sin^2\beta\cos(2\gamma)\Bigg]\Bigg]^2 \frac{1}{16} |c_2^* c_4|^2 \\
& + \left[-\frac{\gamma_e}{\hbar}\left[\delta D \frac{3}{8}\sin(2\beta) - \delta D^{(2)} \frac{1}{4}\sin(2\beta)\cos(2\gamma)\right]\right]^2 \frac{1}{4} |c_1^* c_4 \\
& + c_2^* c_3|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D \frac{3}{8}\sin^2 \\
& - \delta D^{(2)} \frac{1}{4}(1+\cos^2\beta)\cos(2\gamma)\Bigg]\Bigg]^2 |c_1^* c_3|^2 \Bigg\}
\end{aligned}
\tag{6.52}
$$

$$
\begin{aligned}
J_{bbcc}(\omega_{bc}) = \frac{\tau_c}{1+(\omega_{bc}\tau_c)^2} \Bigg\{ & \left[\delta F \frac{3}{8}\sin(2\beta) - \delta F^{(2)} \frac{1}{4}\sin(2\beta)\cos(2\gamma)\right]^2 m_2^2 \\
& + \left(-\frac{\gamma_e}{\hbar} a\right)^2 \frac{1}{4} |c_1^* c_3|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D \frac{1}{2}(3\cos^2\beta - 1) \\
& + \delta D^{(2)} \sin^2\beta\cos(2\gamma)\Bigg]\Bigg]^2 \frac{1}{16} |c_1^* c_3|^2 \\
& + \left[-\frac{\gamma_e}{\hbar}\left[\delta D \frac{3}{8}\sin(2\beta) - \delta D^{(2)} \frac{1}{4}\sin(2\beta)\cos(2\gamma)\right]\right]^2 \frac{1}{4} |c_1^* c_4 \\
& + c_2^* c_3|^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D \frac{3}{8}\sin^2 \\
& - \delta D^{(2)} \frac{1}{4}(1+\cos^2\beta)\cos(2\gamma)\Bigg]\Bigg]^2 |c_2^* c_4|^2 \Bigg\}
\end{aligned}
\tag{6.53}
$$

$$
\begin{aligned}
J_{aaaa}(0) = \tau_0 \Bigg\{ & \left(\frac{\beta_e\, B_0}{\hbar}\delta g\right)^2 \frac{1}{4} \\
& + \left[\delta F \frac{1}{2}(3\cos^2\beta - 1) + \delta F^{(2)} \sin^2\beta\cos(2\gamma)\right]^2 \frac{1}{4} \\
& + \left(-\frac{\gamma_e}{\hbar} a\right)^2 \frac{1}{16} (|c_1|^2 - |c_2|^2)^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D \frac{1}{2}(3\cos^2\beta - 1) \\
& + \delta D^{(2)} \sin^2\beta\cos(2\gamma)\Bigg]\Bigg]^2 \frac{1}{16} (|c_1|^2 - |c_2|^2)^2 \\
& + \Bigg[-\frac{\gamma_e}{\hbar}\Bigg[\delta D \frac{3}{8}\sin(2\beta) \\
& - \delta D^{(2)} \frac{1}{4}\sin(2\beta)\cos(2\gamma)\Bigg]\Bigg]^2 [Re(c_1^* c_2)]^2 \Bigg\}
\end{aligned}
\tag{6.54}
$$

$J_{bbbb}(0) = J_{aaaa}(0)$, and for all indices other than these

$$J_{\alpha\beta\alpha'\beta'}\left(\omega_{\eta\xi}\right) = 0 \tag{6.55}$$

Using Eqs. (6.27), (6.49) to (6.55), one can calculate the elements of the relaxation matrix.

One obtains the non-zero elements of the relaxation matrix, $\hat{\hat{\mathrm{R}}}_{\alpha\alpha'\beta\beta'}$, corresponding to the eigenstates $|a\rangle, |b\rangle, |c\rangle, |d\rangle$ of the static Hamiltonian, $\widehat{H}_0$, for the S = I= ½ coupled system described as follows:

$$\begin{gathered}\hat{\hat{R}}_{aabb} = 2\, J_{abab}(\omega_{ab}); \hat{\hat{R}}_{aacc} = 2\, J_{acac}(\omega_{ac}); \hat{\hat{R}}_{aadd} \\ = 2\, J_{adad}(\omega_{ad})\end{gathered} \tag{6.56}$$

$$\hat{\hat{R}}_{bbaa} = \hat{\hat{R}}_{aabb}; \hat{\hat{R}}_{bbcc} = 2\, J_{bcbc}(\omega_{bc}); \hat{\hat{R}}_{bbdd} = 2\, J_{bdbd}(\omega_{bd}) \tag{6.57}$$

$$\hat{\hat{R}}_{ccaa} = \hat{\hat{R}}_{aacc}; \hat{\hat{R}}_{ccbb} = \hat{\hat{R}}_{bbcc}; \hat{\hat{R}}_{ccdd} = 2\, J_{cdcd}(\omega_{cd}) \tag{6.58}$$

$$\hat{\hat{R}}_{ddaa} = \hat{\hat{R}}_{aadd}; \hat{\hat{R}}_{ddbb} = \hat{\hat{R}}_{bbdd}; \hat{\hat{R}}_{ddcc} = \hat{\hat{R}}_{ccdd} \tag{6.59}$$

$$\begin{gathered}\hat{\hat{R}}_{aaaa} = -\left(\hat{\hat{R}}_{aabb} + \hat{\hat{R}}_{aacc} + \hat{\hat{R}}_{aadd}\right); \hat{\hat{R}}_{bbbb} \\ = -\left(\hat{\hat{R}}_{bbaa} + \hat{\hat{R}}_{bbcc} + \hat{\hat{R}}_{bbdd}\right) \\ \hat{\hat{R}}_{cccc} = -\left(\hat{\hat{R}}_{ccaa} + \hat{\hat{R}}_{ccbb} + \hat{\hat{R}}_{ccdd}\right); \hat{\hat{R}}_{dddd} \\ = -\left(\hat{\hat{R}}_{ddaa} + \hat{\hat{R}}_{ddbb} + \hat{\hat{R}}_{ddcc}\right)\end{gathered} \tag{6.60}$$

$$\begin{gathered}\hat{\hat{R}}_{abab} = 2J_{aabb}(0) - J_{aaaa}(0) - J_{bbbb}(0) + \left(\hat{\hat{R}}_{aaaa} + \hat{\hat{R}}_{bbbb}\right)/2\,; \\ \hat{\hat{R}}_{acac} = 2J_{aacc}(0) - J_{aaaa}(0) - J_{cccc}(0) + \left(\hat{\hat{R}}_{aaaa} + \hat{\hat{R}}_{cccc}\right)/2\,; \\ \hat{\hat{R}}_{adad} = 2J_{aadd}(0) - J_{aaaa}(0) - J_{dddd}(0) + \left(\hat{\hat{R}}_{aaaa} + \hat{\hat{R}}_{dddd}\right)/2\end{gathered} \tag{6.61}$$

$$\begin{gathered}\hat{\hat{R}}_{baba} = \hat{\hat{R}}_{abab} \\ \hat{\hat{R}}_{bcbc} = 2J_{bbcc}(0) - J_{bbbb}(0) - J_{cccc}(0) + \left(\hat{\hat{R}}_{bbbb} + \hat{\hat{R}}_{cccc}\right)/2 \\ \hat{\hat{R}}_{bdbd} = 2J_{bbdd}(0) - J_{bbbb}(0) - J_{dddd}(0) + \left(\hat{\hat{R}}_{bbbb} + \hat{\hat{R}}_{dddd}\right)/2\end{gathered} \tag{6.62}$$

$$\begin{gathered}\hat{\hat{R}}_{caca} = \hat{\hat{R}}_{acac}; \hat{\hat{R}}_{cbcb} = \hat{\hat{R}}_{bcbc} \\ \hat{\hat{R}}_{cdcd} = 2J_{ccdd}(0) - J_{cccc}(0) - J_{dddd}(0) + \left(\hat{\hat{R}}_{cccc} + \hat{\hat{R}}_{dddd}\right)/2\end{gathered} \tag{6.63}$$

$$\hat{\hat{R}}_{dada} = \hat{\hat{R}}_{adad}; \hat{\hat{R}}_{dbdb} = \hat{\hat{R}}_{bdbd}; \hat{\hat{R}}_{dcdc} = \hat{\hat{R}}_{cdcd} \tag{6.64}$$

In the absence of any experimental values available for fluctuations of the various SHPs (defined in Eqs. 4.2-4.11) the same percentage, i.e. $\Delta(SHP)\%$, of the amplitude of each spin Hamiltonian parameters, has been used for calculation of the effect of fluctuation, for simplification. Specifically, the fluctuations of the parameters as defined in Appendix F :

$$\begin{aligned}\delta g &= \Delta(SHP)\%. g,\\ \delta a &= \Delta(SHP)\%. a,\\ \delta F &= \Delta(SHP)\%. F,\\ \delta F^{(2)} &= \Delta(SHP)\%. F^{(2)},\\ \delta D &= \Delta(SHP)\%. D,\\ \delta D^{(2)} &= \Delta(SHP)\%. D^{(2)}\end{aligned} \tag{6.65}$$

6.4 Estimation of correlation time (τ_c)

As for the value of the correlation time, τ_c, required in Eq. (6.36), it has been estimated here in two ways

(i) Using Slichter's Eq. (5.301) (Slichter, 2013):

This formula is:

$$\frac{1}{T_1} = 2\gamma_n^2 \frac{f_p^2}{3} \frac{\tau_c}{1 + (\omega_0\tau_c)^2} \tag{6.66}$$

$$T_1 = \frac{3\omega_0}{\gamma_n^2 f_p^2}$$

where T_1 is minimum when $\omega_0\tau_c = 1$, where

$$\omega_0 = \hbar^{-1}\bar{g}_s\beta_e B_0 \tag{6.67}$$

is the Larmor frequency.

In Eqs. (6.66) and (6.67) the nuclear gyromagnetic ratio $\gamma_n = 42.57\ MHz \cdot T^{-1}$, the Bohr magneton $\beta_e = 9.27400968(20) \times 10^{-24} J \cdot T^{-1}$, the reduced Planck constant $\hbar = 1.054571726(47) \times 10^{-34} J \cdot s$, $B_0 = 3000.0$ Gauss is the intensity of the external magnetic field, $\bar{g}_s$ is the average g factor of the electron spin (Lee *et al*. 1993) . Using now the diagonal matrix $\tilde{g} = (g_{xx}, g_{yy}, g_{zz}) = (2.0026, 2.0035, 2.0033)$ [Lee et al. (1993)], so that $\bar{g}_s = \frac{1}{3}(g_{xx} + g_{yy} + g_{zz}) = 2.0031$, one calculates the value of τ_c when T_1 is minimum from Eq. (6.66) to be:

$$\tau_c = \frac{1}{\hbar^{-1}\bar{g}_s\beta_e B_0} = 1.42 \times 10^{-8} s \tag{6.68}$$

(ii) Use of the relation given by Freed (1965a):

This relation is

$$\frac{0.4 \times 10^6}{\omega_0^2 \tau_c^2} = 30 \times 10^2 sec^{-1} \tag{6.69}$$

where

$$\omega_0 = \hbar^{-1} \bar{g}_s \beta_e B_0 \tag{6.70}$$

so that using the required values as presented in the preceding section

$$\tau_c = \frac{20}{\sqrt{3}\omega_0} = 1.65 \times 10^{-7} s \tag{6.71}$$

The two values as estimated in (i) and (ii) are only order-of-magnitude estimates, leading to the values of τ_c anywhere from $10^{-8}s$ - $10^{-6}s$ based on the assumptions used by Slichter (2013) and Freed (1965a). Furthermore, these are for guidance only; the actual values of τ_c may be outside of this range as no realistic models, other than the two used here, are available to estimate them.

6.5 Simulation of the relaxation as caused by fluctuation of spin-Hamiltonian parameters

Fourier transforms of SECSY signals were simulated, corresponding to the experiments of Lee *et al*. (1993), at the orientations $(\alpha, \beta, \gamma) = (0°, -5°, 0°)$ and $(\alpha, \beta, \gamma) = (0°, 5°, 90°)$, of B_0 with respect to the crystal axes as shown in Figs. 6.1 and 6.2, respectively. As for echo-ELDOR signals, its Fourier transforms were simulated at the orientations $(\alpha, \beta, \gamma) = (0°, -30°, 0°)$ with the mixing times $T_m = 40\mu s$. In particular, for both cases, the simulations were carried out for several choices of correlation times: $\tau_c = 10^{-5}, 10^{-6}, 10^{-7}$ and fluctuation percentage $\Delta(SHP)$ of SHP: $\Delta(SHP) = 10\ \%, 1\%, 0.1\%$ as required in the calculation of matrix elements of the relaxation superoperator.

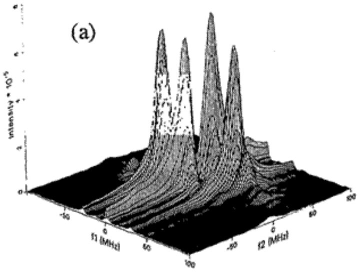
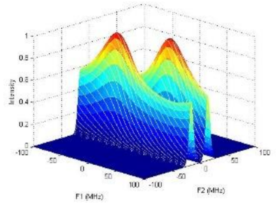
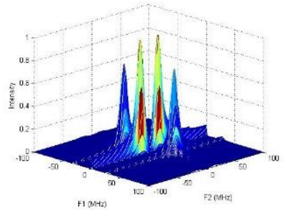
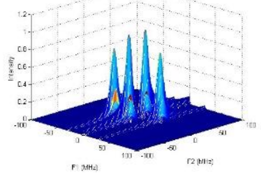
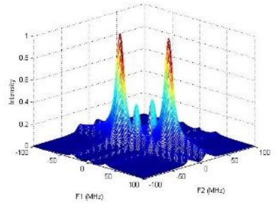
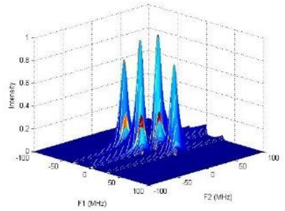
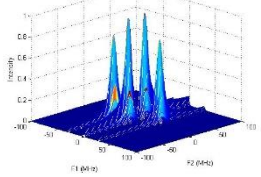
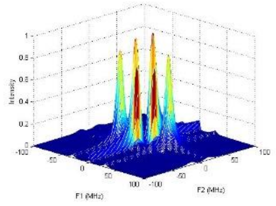
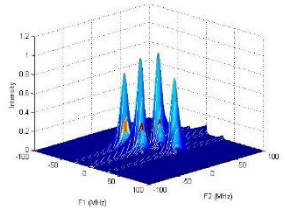
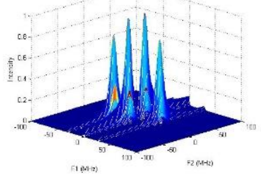
Expriment FT spectra			
Fluctuation amplitude of SHP of the respective SHP	$\tau_c = 1.0 \times 10^{-5} s$	$\tau_c = 1.0 \times 10^{-6} s$	$\tau_c = 1.0 \times 10^{-7} s$
$\Delta(SHP) = 10\%$			
$\Delta(SHP) = 1\%$			
$\Delta(SHP) = 0.1\%$			

Figure 6.1 Simulated Fourier transforms of SECSY signals with relaxation due to fluctuation signals of Lee *et al.* (1993) Figure 8 (reproduced here with the permission of the authors).

The experimental Figure reproduced by permission from Figure 8 of Lee *et al*. (1993) (upper). Simulation with relaxation due to fluctuation, $\Delta(SHP)$, taken into account (lower rows) SECSY spectra for various values of time ($\tau_c = 1.0 \times 10^{-5} s - 1.0 \times 10^{-7} s$) at the orientations $(\alpha, \beta, \gamma) = (0°, -5°, 0°)$. An inhomogeneous broadening width $\Delta = 4MHz$ is used in the simulation.

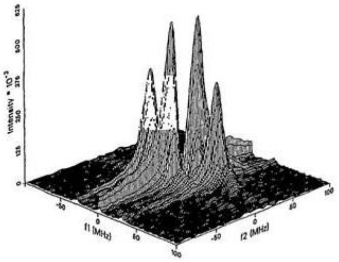
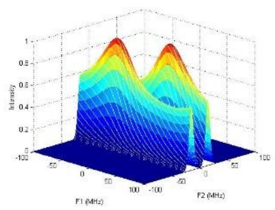
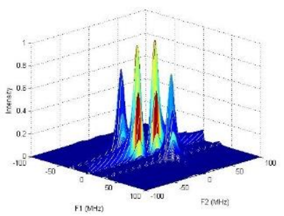
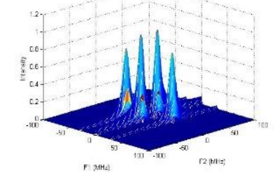
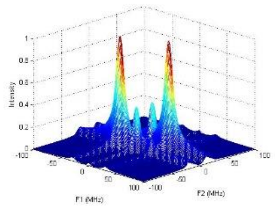
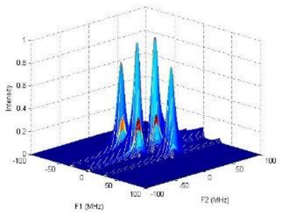
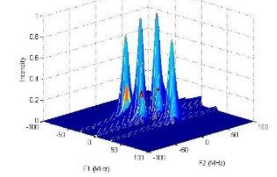
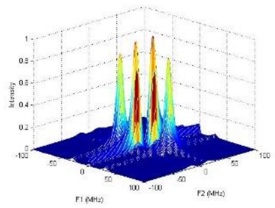
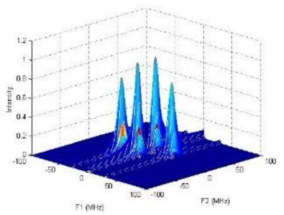
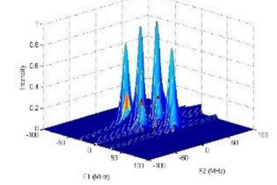
Experiment FT spectrum [Lee *et al*. (1993)]			
Fluctuation amplitude of SHP of the respective SHP	$\tau_c = 1.0 \times 10^{-5} s$	$\tau_c = 1.0 \times 10^{-6} s$	$\tau_c = 1.0 \times 10^{-7} s$
$\Delta(SHP) = 10\%$			
$\Delta(SHP) = 1\%$			
$\Delta(SHP) = 0.1\%$			

Figure 6.2 Fourier transforms of simulated SECSY signals with relaxation due to fluctuation and Figure 9a of Lee *et al.* (1993) (reproduced here with the permission of the authors.) Simulations of SECSY spectra with relaxation due to fluctuation, $\Delta(SHP)$, taken into account (lower row) for various values of correlation time ($\tau_c = 1.0 \times 10^{-5}s - 1.0 \times 10^{-7}s$) at the orientations $(\alpha, \beta, \gamma) = (0°, 5°, 90°)$. An inhomogeneous broadening width $\Delta = 4MHz$ is used in the simulation.

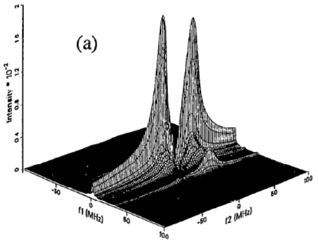
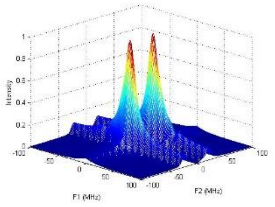
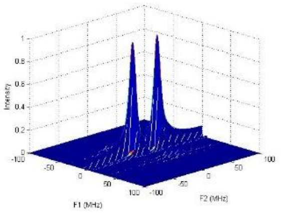
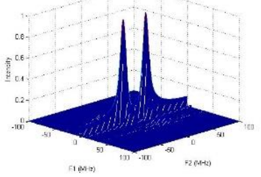
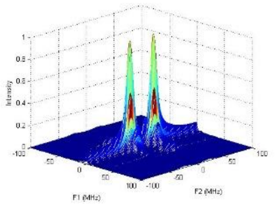
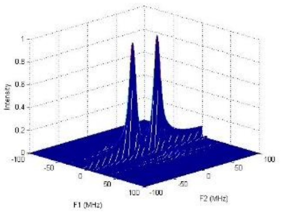
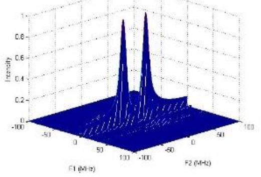
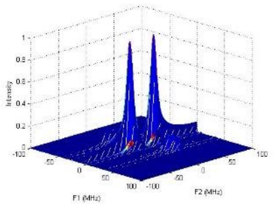
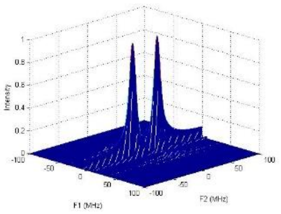
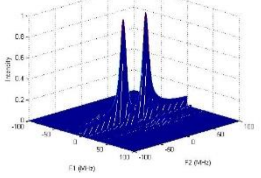
Expriment FT spectra			
Fluctuation amplitude of SHP of the respective SHP	$\tau_c = 1.0 \times 10^{-5} s$	$\tau_c = 1.0 \times 10^{-6} s$	$\tau_c = 1.0 \times 10^{-7} s$
$\Delta(SHP) = 10\%$			
$\Delta(SHP) = 1\%$			
$\Delta(SHP) = 0.1\%$			

Figure 6.3 The Fourier transforms of simulated SECSY signals with relaxation due to fluctuation and Figure 11a of Lee *et al.* (1993) (reproduced here with the permission of the authors.) Simulations of SECSY spectra with relaxation due to fluctuation, $\Delta(SHP)$, taken into account (lower row) for various values of correlation time ($\tau_c = 1.0 \times 10^{-5} s - 1.0 \times 10^{-7} s$) at the orientations $(\alpha, \beta, \gamma) = (0°, -30°, 0°)$. An inhomogeneous broadening width $\Delta = 4MHz$ is used in the simulation

It is seen that for the two cases considered (SECSY and echo-ELDOR) the best agreement is obtained for the following ranges: $\tau_c = 10^{-6}s$ - $10^{-7}s$ and $\Delta(SHP) = 0.1$ % - 1.0 %. The value of $\tau_c \sim 10^{-7}s$ as calculated by Freed (1965) is found to be in the range of values found here that provide a good agreement of the calculated spectra with the experimental ones. It is noted that fluctuation by the same percentage is assumed here for all SHPs, whereas in reality they may vary individually by different amounts. If the realistic individual fluctuations were taken into account, along with more realistic τ_c values than considered here, one may obtain a much better agreement of theoretical simulations with the experimental spectra. In the light of these observations, it appears that the fluctuation model considered here, may indeed reproduce the experimental relaxation data to a satisfactory agreement.

7. Concluding remarks and future perspectives

Concluding remarks. This thesis is devoted to the subject of computations for pulse EPR by solving the Liouville von Neumann equation in Liouville space in the presence of relaxation, providing a comprehensive theoretical treatment, including a detailed description of relaxation processes.

The *conclusions* of the present work are as follows:

(i) An algorithm for simulation of pulse EPR signals taking into account relaxation rigorously by the use of LVN equation, based on solving the relevant matrix differential equation in Liouville space, has been developed. The algorithm is illustrated here to simulate SECSY and echo-ELDOR signals for an electron-nuclear spin coupled system (electron spin S=1/2; nuclear spin I=1/2) in an irradiated malonic-acid crystal, taking into account spin-phonon modulation. The simulated results so obtained are compared with the experimental results of Lee *et al.* (1993). The experimental and simulated spectra are found to be in reasonably good agreement.

(ii) The developed algorithm is used in addition to simulate the spectra for a specific case of the electron-electron dipolar interaction of two nitroxide radicals, each with electron spin S=1/2, nuclear spin I=1/2, for a proposed experiment, intended for distance measurements. The Pake doublets in the Fourier transform of both the signals calculated with and without relaxation correctly indicate the spacing of 10 MHz of dipolar coupling as assumed in the simulation. Furthermore, inclusion of the relaxation is found to produce a significant broadening of the signal as seen in the Fourier transform.

(iii) This algorithm is also extended to calculate the effect of relaxation due to fluctuation of spin-Hamiltonian parameters by phonon modulation (Freed and Fraenkel, 1963; Slichter, 2013). From a comparison of the Figures showing the simulated SECSY and echo-ELDOR signals for an electron-nuclear spin-coupled system (S = I = ½) in an irradiated malonic acid crystal as calculated for $\tau_c = 10^{-5}, 10^{-6}, 10^{-7}$ and the fluctuation $\Delta(SHP) = 10\ \%, 1\%, 0.1\%$ of their static values as fitting parameters with the experimental results of Lee *et al.* (1993), it is seen that for the two cases considered (SECSY and echo-ELDOR) the best agreement is obtained for the following ranges: $\tau_c = 10^{-6}s$ - $10^{-7}s$ and $\Delta(SHP) = 0.1$ % - 1.0 %. The value of $\tau_c \sim 10^{-7}s$ as calculated by Freed (1965) is found to be in the range of values found here that provide a good agreement of the calculated spectra with the experimental ones. It is noted that fluctuation by the same percentage is assumed here for all SHPs, whereas in reality they may vary individually by different amounts. If the realistic individual fluctuations were taken into account, along with more realistic τ_c values than considered here, one may obtain a much better agreement of theoretical simulations with the experimental spectra. In the light of these observations, it appears that the fluctuation model considered here, may indeed be refined further to reproduce the simulations to be in satisfactory agreement with the experiment.

Future perspectives. The following two problems, which are natural extensions of the problems investigated in this thesis, are worthy of investigation.

1. Fluctuation of individual parameters in the calculation of relaxation. In this thesis was investigated relaxation of nitroxide radicals with electron spin S=1/2, nuclear spin I=1/2 in malonic acid crystal due to fluctuations of SHP caused by spin-phonon interaction. However, all spin-Hamiltonian parameters (SHP) were assumed to be undergoing fluctuation by the same percentage of their static values. In reality, different SHPs fluctuate at different rates. Hence, a more precise calculation should take into account realistic values of fluctuations for different SHPs. For examples, a model was proposed by Fresch *et al.* (2006) for such a calculation, which can be exploited to this end.

2. The calculation of a powder signal for coupled nitroxides taking into account relaxation using the simulation technique developed in this thesis, extending the calculations of Misra et al. (2009) (for a pair of dipolar-interaction coupled nitroxide radicals, each with electron spin S =1/2 and nuclear spin I=1, not including relaxation), should be carried out similar to the one done here for a pair of coupled nitroxide radicals, each with S=1/2 and I=1/2. This will require simulations in Liouville space of dimension $((2S+1)x(2I+1))$ x $((2S+1)x(2I+1))^2$ = $(36x36)^2$, a rather large one! However, sophisticated numerical techniques and faster computers might achieve this goal in the not-too-distant future.

References

Abragam, Anatole. 1961. *The principles of nuclear magnetism.* Vol. No. 32. Oxford university press.

Åman, Ken, and Per-Olof Westlund. 2007. "Direct calculation of 1 H 2 OT 1 NMRD profiles and EPR lineshapes for the electron spin quantum numbers S= 1, 3/2, 2, 5/2, 3, 7/2, based on the stochastic Liouville equation combined with Brownian dynamics simulation." *Physical Chemistry Chemical Physics* 9 (6): 691-700.

Atta-Ur-Rahman, Muhammad Iqbal Choudhary, Atia-tul- Wahab. 2015. "2.5.1 Phase Cycling and Coherence Transfer Pathways." In *Solving Problems with NMR Spectroscopy*. Elsevier Science Publishing Co Inc.

Bain, Alex D. 1984. "Coherence levels and coherence pathways in NMR. A simple way to design phase cycling procedures." *Journal of Magnetic Resonance (1969)* 56 (3): 418-427.

Blume, R. J. 1958. "Electron Spin Relaxation Times in Sodium-Ammonia Solutions." *Physical Review* 109: 1867–1873.

Bodenhausen, G., H. Kogler, and RR Ernst. 1984. "Selection of coherence-transfer pathways in NMR pulse experiments." *Journal of Magnetic Resonance (1969)* 58 (3): 370-388.

Born, M. 1926. "Quantenmechanik der stoßvorgange." *Z. Physik A Hadrons and Nuclei* 38(11): 38(11):803–827.

Dan Gamliel, and Jack H. Freed, a. 1990. "Theory of two-dimensional ESR with nuclear modulation." *Journal of Magnetic Resonance (1969)* 89 (1): 60-93.

Deligiannakis, Yiannis, Maria Louloudi, and Nick Hadjiliadis. 2000. "Electron spin echo envelope modulation (ESEEM) spectroscopy as a tool to investigate the coordination environment of metal centers." *Coordination Chemistry Reviews* 204 (1): 1-112.

Dirac, Paul Adrien Maurice. 1981. *The principles of quantum mechanics.* Vol. No. 27. Oxford university press.

EPR_wikipedia. 2019. *Pulsed electron paramagnetic resonance.* Accessed 2019. https://en.wikipedia.org/wiki/Pulsed_electron_paramagnetic_resonance.

Franck, John M., Siddarth Chandrasekaran, Boris Dzikovski, Curt R. Dunnam, and Jack H. Freed. 2015. "Focus: Two-dimensional electron-electron double resonance and molecular motions: The challenge of higher frequencies." *The Journal of chemical physics* 142 (21): 212302.

Freed, Jack H. 1965. "Theory of Saturation and Double-Resonance Effects in ESR Spectra." *The Journal of Chemical Physics* 43 (7): 2312-2332.

Freed, Jack H. 1965a. "Quantum Effects of Methyl-Group Rotations in Magnetic Resonance: ESR Splittings and Linewidths." *The Journal of chemical physics* 43 (5): 1710-1720.

Freed, Jack H. 1979. "Theory of multiple resonance and ESR saturation in liquids and related media." In *Multiple Electron Resonance Spectroscopy*, 73-142. Springer.

Freed, Jack H., and George K. Fraenkel. 1963. "Theory of linewidths in electron spin resonance spectra." *The Journal of Chemical Physics* 39 (2): 326-348.

Freed, Jack H.. 1968. "Generalized Cumulant Expansions and Spin-Relaxation Theory." *The Journal of chemical physics* 49 (1): 376-391.

Fresch, B., Frezzato, D., Moro, G. J., Kothe, G., & Freed, J. H. 2006. "Collective Fluctuations in Ordered Fluids Investigated by Two-Dimensional Electron− Electron Double Resonance Spectroscopy." *The Journal of Physical Chemistry B* 110 (47): 24238-24254.

G. Bachman, L. Naricci. 1966. *Functional Analysis.* Academic Press.

Gamliel, Dan, and Haim Levanon. 1995. *Stochastic processes in magnetic resonance.* World Scientific.

García-Rubio, Inés, Alexander Angerhofer, and Arthur Schweiger. 2007. "EPR and HYSCORE investigation of the electronic structure of the model complex Mn (imidazole) 6: Exploring Mn (II)–imidazole binding using single crystals." *J. Magn. Reson.* 184 (1): 130.

Gemperle, C., G. Aebli, A. Schweiger, and R. R. Ernst. 1990. "Phase cycling in pulse EPR." *Journal of Magnetic Resonance (1969)* 88 (2): 241-256.

Gordon, J. P., and K. D Bowers. 1958. "Microwave Spin Echoes from Donor Electrons in Silicon." *Physical Review Letters* 1: 368–370.

Hahn, E.L. 1950. "Spin echoes." *Physical Review* (80): 580–594.

Håkansson, Pär, ThaoNguyen Nguyen, Prasanth B. Nair, Ruth Edge, and Eugen Stulz. 2013. "Cu (ii)–porphyrin molecular dynamics as seen in a novel EPR/Stochastic Liouville equation study." *Physical Chemistry Chemical Physics* 15 (26): 10930-10941.

Hanson, Graeme R., Kevin E. Gates, Christopher J. Noble, Mark Griffin, Anthony Mitchell, and Simon Benson. 2004. "XSophe-Sophe-XeprView®. A computer simulation software suite (v. 1.1. 3) for the analysis of continuous wave EPR spectra." *J. Inorg. Biochem.* 98 (5): 903-916.

Hogben, H. J., M. Krzystyniak, G. T. P. Charnock, P. J. Hore, and Ilya Kuprov. 2011. "Spinach–a software library for simulation of spin dynamics in large spin systems." 208 (2): 179-194.

Jeener, Jean. 1982. "Superoperators in magnetic resonance." *Advances in Magnetic and Optical Resonance.* Academic Press. 1-51.

Landau, Lev Davidovich, and Evgenii Mikhailovich Lifshitz. 2013. *Quantum mechanics: non-relativistic theory.* Vol. 3. Elsevier.

Lee, Sanghyuk, Baldev R. Patyal, and Jack H. Freed. 1993. "A two-dimensional Fourier transform electron-spin resonance (ESR) study of nuclear modulation and spin relaxation in irradiated malonic acid." *The Journal of chemical physics* 98 (5): 3665-3689.

Libertini, Louis J., and O. Hayes Griffith. 1970. "Orientation Dependence of the Electron Spin Resonance Spectrum of Di-t-butyl Nitroxide." *The Journal of Chemical Physics* 53 (4): 1359-1367.

Magnus, Jan R., and Heinz Neudecker. 1995. *Matrix differential calculus with applications in statistics and econometrics.* Wiley & Sons.

McConnell, H. M., C. Heller, T. Cole, and R. W. Fessenden. 1960. "Radiation damage in organic crystals. I. CH (COOH) 2 in malonic acid1." *Journal of the American Chemical Society* 82 (4): 766-775.

Merzbacher, Eugen. 1998. *Quantum Mechanics.* Wiley, John & Sons, Inc.

Mims, W. B. 1965. "Pulsed endor experiments." *Proceedings of the Royal Society A.* 52–457.

Mims, W. B., K. Nassau, and J. D. McGee. 1961. "Spectral Diffusion in Electron Resonance Lines." *Physical Review* 123: 2059–2069.

Misra, Sushil K. 2011. *Multifrequency Electron Paramagnetic Resonance: Theory and Applications.* John Wiley & Sons.

Misra, Sushil K. and Jack H. Freed. 2011a. "Distance measurements: continuous-wave (CW)- and pulsed dipolar EPR." In *Multifrequency Electron Paramagnetic Resonance: Theory & Applications,* by S. K. Misra, 545-588. Wiley-VCH, Weinheim.

Misra, Sushil K. and Lin Li. 2015. "Spin-Hamiltonian Parameters (SHP) of a Gd 3+-Doped Y (BrO 3) 3• 9H 2 O Single Crystal as Studied by Electron Paramagnetic Resonance at 110 and 300 K: a Comparison with SHPs in Other R (BrO 3) 3• 9H 2 O [(R= Pr, Nd, Sm, Eu, Dy)] Crystals." *Applied Magnetic Resonance* 46 (10): 1069-1077.

Misra, Sushil K. and Lin Li. 2018. "A Rigorous Procedure for Calculation of Pulsed EPR Signal with Relaxation." *J. Apl. Theol* 2 (1): 5-16.

Misra, Sushil K., Lin Li, Sudip Mukherjee, and Goutam Ghosh. 2015. "Anisotropic magnetic field observed at 300 K in citrate-coated iron oxide nanoparticles: effect of counterions." *Journal of Nanoparticle Research* 17 (12): 487.

Misra, Sushil K., Peter P. Borbat, and Jack H. Freed. 2009. "Calculation of double-quantum-coherence two-dimensional spectra: distance measurements and orientational correlations." *Applied magnetic resonance* 36 (2-4): 237-258.

Pribitzer, Stephan, Andrin Doll, and Gunnar Jeschke. 2016. "SPIDYAN, a MATLAB library for simulating pulse EPR experiments with arbitrary waveform excitation." *Journal of Magnetic Resonance* 263: 45-54.

Prisner, Thomas, Martin Rohrer, and Fraser MacMillan. 2001. "Pulsed EPR spectroscopy: biological applications." *Annual review of physical chemistry* 52 (1): 279-313.

Redfield, Alfred G. 1957. "On the theory of relaxation processes." *IBM Journal of Research and Development* 1 (1): 19-37.

Schwartz, Leslie J., Arthur E. Stillman, and Jack H. Freed. 1982. "Analysis of electron spin echoes by spectral representation of the stochastic Liouville equation." *J. Chem. Phys.* 77 (11): 5410-5425.

Schweiger, A. & Jeschke. 2001. *Principles of pulse electron paramagnetic resonance.* New York: Oxford University Press.

Slichter, Charles P. 2013. *Principles of magnetic resonance.* Vol. 1. Springer Science & Business Media.

Stoll, Stefan and R. David Britt. 2009. "General and efficient simulation of pulse EPR spectra." *Phys. Chem. Chem. Phys.* 11 (31): 6614-6625.

Stoll, Stefan, and Arthur Schweiger. 2006. "EasySpin, a comprehensive software package for spectral simulation and analysis in EPR." *J. Magn. Reson.* 178 (1): 42-55.

Stoll, Stefan, and B. Kasumaj. 2008. "Phase cycling in electron spin echo envelope modulation." *Applied magnetic resonance* 35 (1): 15-32.

Van Doorslaer, Sabine, and Evi Vinck. 2007. "The strength of EPR and ENDOR techniques in revealing structure–function relationships in metalloproteins." *Physical Chemistry Chemical Physics* 9 (33): 4620-4638.

Vinck, E., and S. Van Doorslaer. 2004. "Analysing low-spin ferric complexes using pulse EPR techniques: a structure determination of bis (4-methylimidazole)(tetraphenylporphyrinato) iron (III)." *Phys. Chem. Chem. Phys.* 6 (23): 5324-5330

Appendix A. Reprint of the publication: Misra, Sushil K. and Lin Li, a. 2018. "A Rigorous Procedure for Calculation of Pulsed EPR Signal with Relaxation." *J. Apl. Theol* 2 (1): 5-16. [It is noted that there were errors in the factors multiplying the square brackets in Eqs. (A.6) and (A.7) for the coefficients A and B, respectively, in this reprint as they were quoted *verbatim* from Lee *et al.* (1993). The correct expressions are used in this thesis, as seen in Eqs. (4.3) and (4.4), respectively, for the coefficients A and B.] {N.B. The experimental Figures reproduced in this paper for comparison have the permission of the authors of Lee *et al.* (1993)]}

VERIZONA PUBLISHER
Publishing Your Ideas

www.verizonaonlinepublishing.com
Vol: 2, Issue: 1

Journal of Applied and Theoretical Physics Research

A Rigorous Procedure for Calculation of Pulsed EPR Signals with Relaxation

Sushil K. Misra* and Lin Li

Physics Department, Concordia University, 1455 de Maisonneuve Boulevard West, Montreal, Quebec H3G 1M8, Canada

*Corresponding author: Sushil K. Misra, Physics Department, Concordia University, 1455 de Maisonneuve Boulevard West, Montreal, Quebec H3G 1M8, Canada; E mail: sushil.misra@concordia.ca

Article Type: Research, Submission Date: 23 December 2017, Accepted Date: 16 January 2018, Published Date: 20 February 2018.

Citation: Sushil K. Misra and Lin Li (2018) A Rigorous Procedure for Calculation of Pulsed EPR Signals with Relaxation. J Apl Theol 2(1): 5-16.

Abstract

A procedure is developed to calculate pulsed electron paramagnetic resonance (EPR) signals with relaxation rigorously. It can be carried out within a reasonable time on a PC using Fortran or Matlab, not requiring any sophisticated software. The flow chart for this kind of simulation is included. It is illustrated here numerically, as coded in Matlab, to calculate the two-dimensional spin echo correlation spectroscopy (2D-SECSY) and two-dimensional electron-electron double-resonance (2D-ELDOR) signals for two cases: (i) a coupled electron-nuclear system with the electron spin (S= ½) and nuclear spin (I= ½) corresponding to the experimental results of Lee, Patyal and Freed [1] in a malonic acid single crystal, and (ii) 2D-ELDOR signal obtained for an electron-electron dipolar-coupled system of two nitroxide radicals in a malonic acid single crystal, with hyperfine interactions included, for a proposed experiment, intended for distance measurements, important for biological systems. The limits of applicability of the algorithm used here are discussed.

Keywords: Pulsed EPR, Two-dimensional spin-echo-correlation spectroscopy (2D-SECSY), Two-dimensional electron-electron double-resonance (2D-ELDOR), Stochastic Liouville equation (SLE), Electron-electron dipolar-coupled system, Distance measurements in biological systems.

Introduction

Pulsed electron paramagneticresonance (EPR) is a powerful technique that can be exploited to reveal the electronic and geometric structures in detail around paramagnetic centers, including nanoscale environments. It has a wide range of applications in biology, chemistry, physics and materials science [1–5]. It can distinguish weak interactions between electron spins, as well as those between electron and nuclear spins, not resolved by continuous wave (CW) EPR. A rigorous simulation of pulsed EPR spectra is necessary for a quantitative analysis to extract correctly information on electronic and geometric structure from pulsed EPR data.

This paper is devoted to the very important subject of pulsed EPR in the presence of relaxation. In most cases, relaxation processes are ignored because of high conceptual and computational complexity of a proper description of relaxation processes. This paper provides a comprehensive theoretical treatment including a detailed description of relaxation process.

There are availabletwo open-source packages that are implemented in Matlab, which are much moregeneral than the approach introduced in the present manuscript, namely SPINACH [6] and SPIDYAN [7]. The present procedure, on the other hand, offers an advantage in that within the framework of its applicability it can be carried out on a lap top within a short time using Matlab or Fortran, and does not require any costly software. From this point of view, it is felt that this work serves a useful purpose to practitioners of EPR pulse spectroscopy needing to treat relaxation rigorously in a simple manner.

In order to take into account relaxation rigorously, it is imperative to use Liouvillevon Neumann (LVN) equation, which is an exact quantum-mechanical equation of motion for the density matrix.The Stochastic Liouville Equation (SLE) is a stochastic version of the LVN equation. This equation is valid even for relatively slow random processes, and is therefore especially suitable for EPR, where the natural timescale is short so that the random processes are not usually fast on this time scale. There are three cases, where SLE is applicable:

(a) Time-independent SLE, which does not include any time-dependent Hamiltonians;

(b) Time-dependent SLE, which includes in addition to (a) a

Appendix B. Vectorization of a Matrix

Definition of Vectorization: Let $a_i \in \mathbb{C}^m, i = 1,2,\cdots,n$, denote the columns of $A = [a_1, a_2, \cdots, a_n] \in \mathbb{C}^{m\times n}$. Then the vector $Col(A)$ is defined by Magnus and Neudecker (1995)

$$Col(A) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \in \mathbb{C}^{mn} \tag{B.72}$$

$A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times p}$, $C \in \mathbb{C}^{p\times q}$

$$Col(ABC) = (C^T \otimes A)Col(B) \tag{B.73}$$

$$Col(AB) = \left(I_p \otimes A\right)Col(B) \tag{B.74}$$

$$Col(CA^T) = \left(A \otimes I_p\right)Col(C) \tag{B.75}$$

Properties of the Kronecker Product

$A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{p\times q}$

$$C = A \otimes B \tag{B.76}$$

$$c_{ij,kl} = a_{i,k}b_{j,l} \tag{B.77}$$

let

$$\alpha \equiv p(i-1) + j \tag{B.78}$$

$$\beta \equiv q(k-1) + l \tag{B.79}$$

Then

$$c_{\alpha,\beta} = a_{i,k}b_{j,l} \tag{B.80}$$

Kronecker product has relations to other matrix operations:

$A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{r\times s}$, $C \in \mathbb{C}^{n\times p}$, $D \in \mathbb{C}^{s\times t}$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{B.81}$$

A^T denotes the matrix transpose

A^* denotes the matrix with complex conjugated entries

$A^\dagger$ denotes the conjugate transpose or Hermitian transpose of the matrix.

Appendix C. Equivalence of different ways of expressing Liouville von Neumann (LVN) equation

In this appendix it is shown that the vectorized version of LVN equation as used in this paper is, indeed, equivalent to the matrix form of LVN equation.

(i) The matrix form of Liouville von Neumann equation

LVN equation is described as (Abragam, 1961; Gamliel and Levanon, 1995; Lee *et al*., 1993):

$$\frac{d}{dt}\rho = -i\left[\widehat{H},\rho\right] = -i\left(\widehat{H}\rho - \rho\widehat{H}\right) \tag{C.1}$$

Its matrix form can be calculated as follows:

$$\begin{aligned}\frac{d}{dt}\rho_{ij} &= -i\sum_{k=1}^{k=n}\left(\widehat{H}_{ik}\rho_{kj} - \rho_{ik}\widehat{H}_{kj}\right)\\ &= -i\sum_{k,l=1}^{k=n}\left(\widehat{H}_{ik}\delta_{lj} - \delta_{ik}\widehat{H}_{lj}\right)\rho_{kl}\\ &\equiv -i\sum_{k,l=1}^{k=n}\hat{\widehat{L}}_{ij,kl}\rho_{kl};\ i,j = 1,\cdots,n\end{aligned} \tag{C.2}$$

This equation defines the matrix elements of the Liouville superoperator, or Liouvillian, $\hat{\widehat{L}}$ as expressed in the last term.

(ii) The vectorization of LVN equation

$$\frac{d}{dt}Col(\rho(t)) = Col\left[\widehat{H},\rho(t)\right] \tag{C.3}$$

where

$$\hat{\widehat{L}}\hat{\widehat{\boldsymbol{\rho}}} = \left(I_n \otimes \widehat{H} - \widehat{H}^T \otimes I_n\right)Col(\rho) \tag{C.4}$$

and

$$\hat{\widehat{L}} \equiv \left(I_n \otimes \widehat{H} - \widehat{H}^T \otimes I_n\right) \tag{C.5}$$

and

$$\hat{\hat{\boldsymbol{\rho}}} \equiv Col(\rho(t)) \tag{C.6}$$

(ii) Equivalence of vectorization of LVN equation and the matrix form of LVN equation. From Eqs.(C.3) and (C.4) one can calculate the vectorized form of LVN, Eq. (C.1), as follows:

$$\frac{d}{dt}Col(\rho) = -i\left(I_n \otimes \hat{H} - \hat{H}^T \otimes I_n\right)Col(\rho) \tag{C.7}$$

In Eq. (C.7) it is noted that $\hat{\hat{L}} = \left(I_n \otimes \hat{H} - \hat{H}^T \otimes I_n\right)$, the same as that expressed by Eq.(C.5) in vectorized form.

Let the vectorized form of density matrix ρ be denoted as:

$$\hat{\hat{\boldsymbol{\rho}}} = Col(\rho) \equiv c\rho \tag{C.8}$$

The elements of column $c\rho$ are related to the elements of matrix ρ as follows:

$$\rho_{ij} = c\rho_{i+n(j-1)} \tag{C.9}$$

The matrix elements in of Eq. (C.7) are:

$$\frac{d}{dt}c\rho_{i+n(j-1)} = -i\sum_{q=1}^{q=n\times n}\left[\left(I_n \otimes \hat{H}\right)_{i+n(j-1),q} - \left(\hat{H}^T \otimes I_n\right)_{i+n(j-1),q}\right]c\rho_q \tag{C.10}$$

It is noted that on the right side of Eq.(C.10), in the first term the non-zero elements of the row $i + n(j - 1)$ of the matrix:

$$\left(I_n \otimes \hat{H}\right)_{i+n(j-1),q} = \begin{bmatrix} \hat{H} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \hat{H} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & 0 & \cdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 0 & \hat{H} \end{bmatrix}_{i+n(j-1),q} \tag{C.11}$$

occur in the columns $k + n(j - 1); k = 1, n$ in the j -th $\hat{H}$ matrix block on the diagonal of Eq.(C.11). That is, they are

$$\left(I_n \otimes \hat{H}\right)_{i+n(j-1),k+n(j-1)} = \hat{H}_{ik};\ i,k = 1,\cdots,n \tag{C.12}$$

Introducing now this fact in the first term in Eq.(C.10), one obtains

$$\begin{aligned}&\sum_{q=1}^{q=n\times n}\left[\left(I_n\otimes\widehat{H}\right)_{i+n(j-1),q}\right]c\rho_q\\&=\sum_{k=1}^{k=n}\left[\left(I_n\otimes\widehat{H}\right)_{i+n(j-1),k+n(j-1)}\right]c\rho_{k+n(j-1)}=\sum_{k=1}^{k=n}\widehat{H}_{ik}\rho_{kj}\end{aligned}\tag{C.13}$$

where the final term in Eq.(C.13) is expressed in matrix notation.

As for the second term on the right side of Eq.(C.10), a non-zero element of the matrix:

$$\left(\widehat{H}^T\otimes I_n\right)_{i+n(j-1),q}=\begin{bmatrix}\widehat{H}_{11}^TI_n & \cdots & \widehat{H}_{1i}^TI_n & \cdots & \widehat{H}_{1n}^TI_n\\ \vdots & \ddots & \vdots & \cdots & \vdots\\ \widehat{H}_{j1}^TI_n & \cdots & \widehat{H}_{jj}^TI_n & \cdots & \widehat{H}_{jn}^TI_n\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \widehat{H}_{n1}^TI_n & \cdots & \widehat{H}_{n1}^TI_n & \cdots & \widehat{H}_{nn}^TI_n\end{bmatrix}_{i+n(j-1),q}\tag{C.14}$$

occurs in the diagonal element of a $\widehat{H}_{jk}^TI_n$ block, which is

$$\left(\widehat{H}^T\otimes I_n\right)_{i+n(j-1),i+n(k-1)}=\widehat{H}_{jk}^T\tag{C.15}$$

where $i,j,k=1,\cdots,n$. Thus, in the second term on the right of Eq.(C.10), one finds by retaining only the non-zero terms

$$\begin{aligned}&\sum_{q=1}^{q=n\times n}\left[\left(\widehat{H}^T\otimes I_n\right)_{i+n(j-1),q}\right]c\rho_q\\&=\sum_{k=1}^{k=n}\left[\left(\widehat{H}^T\otimes I_n\right)_{i+n(j-1),i+n(k-1)}\right]c\rho_{i+n(k-1)}\\&=\sum_{k=1}^{k=n}\widehat{H}_{jk}^T\rho_{ik}=\sum_{k=1}^{k=n}\rho_{ik}\widehat{H}_{kj}\end{aligned}\tag{C.16}$$

The non-zero terms on the right of Eq. (C.10) can now be summarized in the matrix form to be as follows:

$$\begin{aligned}&\frac{d}{dt}c\rho_{i+n(j-1)}\\&=-i\sum_{q=1}^{q=nn}\left[\left(I_n\otimes\widehat{H}\right)_{i+n(j-1),q}-\left(\widehat{H}^T\otimes I_n\right)_{i+n(j-1),q}\right]c\rho_q\\&=-i\sum_{k=1}^{k=n}\left(\widehat{H}_{ik}\rho_{kj}-\rho_{ik}\widehat{H}_{kj}\right)=-i\sum_{k,l=1}^{k,l=n}\widehat{\hat{L}}_{ij,kl}\rho_{kl}\end{aligned}\tag{C.17}$$

Thus, comparing with Eq. (A.25), it is seen that vectorization of LVN equation is, indeed, equivalent to its matrix form.

(iii) Expansion in terms of the basis states $|i\rangle\langle j|$

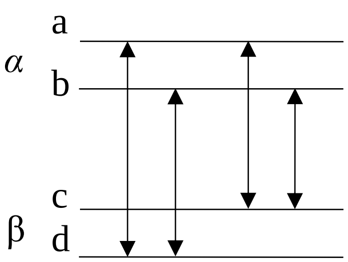
Gamliel and Levanon (1995), and Jeener (1982) also obtained the same result as in Eq.(C.2) by calculating the elements of the Liouvillian as the coefficients in its expansion in terms of the basis states,$|i\rangle\langle j|$, in Liouville space:

$$\begin{aligned}&\hat{\hat{L}}_{ij,kl} = \langle\langle(|i\rangle\langle j|)\big|\hat{H}\otimes I_n - I_n\otimes\hat{H}\big|(|k\rangle\langle l|)\rangle\rangle \\ &= Tr\big[(|i\rangle\langle j|)^\dagger\big(\hat{H}\otimes I_n - I_n\otimes\hat{H}\big)(|k\rangle\langle l|)\big] = \hat{H}_{ik}\delta_{jl} - \delta_{ik}\hat{H}^T_{jl}\end{aligned} \tag{C.18}$$

This is consistent with Eq.(C.2).

Appendix D. Eigenvalues and eigenvectors of a four Level Electron Nuclear System

The four-level Hamiltonian system is shown below, where α, β show the electron Zeeman levels, and the labels a, b, c, d show the hyperfine levels.



The Hamiltonian which operates between pulses is express in a rotating (on-resonance) frame, defined by (Lee *et al.*, 1993)

$$\mathcal{H}_0 = CS_z - \omega_I I_z + AS_z I_z + \frac{1}{2} B S_z I_+ + \frac{1}{2} B^* S_z I_- \tag{D.1}$$

The hyperfine interaction for $I_z = \pm\frac{1}{2}$ is

$$\mathcal{H}_0 = \begin{array}{c} \\ |1\rangle = |++\rangle \\ |2\rangle = |+-\rangle \\ |3\rangle = |-+\rangle \\ |4\rangle = |--\rangle \end{array} \begin{array}{c} \begin{array}{cccc} |1\rangle = |++\rangle & |2\rangle = |+-\rangle & |3\rangle = |-+\rangle & |4\rangle = |--\rangle \end{array} \\ \begin{bmatrix} \frac{C}{2} - \frac{\omega_n}{2} + \frac{A}{4} & \frac{B}{4} & 0 & 0 \\ \frac{B^*}{4} & \frac{C}{2} + \frac{\omega_n}{2} - \frac{A}{4} & 0 & 0 \\ 0 & 0 & -\frac{C}{2} - \frac{\omega_n}{2} - \frac{A}{4} & -\frac{B}{4} \\ 0 & 0 & -\frac{B^*}{4} & -\frac{C}{2} + \frac{\omega_n}{2} + \frac{A}{4} \end{bmatrix} \end{array} \tag{D.2}$$

Let

$$\mathcal{H}_{\alpha,\beta} = \pm\frac{C}{2} - \omega_n I_z \pm \frac{A}{2} I_z \pm \frac{B}{4} I_+ \pm \frac{B^*}{4} I_- \tag{D.3}$$

and states

$|1\rangle = |++\rangle,\ |2\rangle = |+-\rangle$

$|3\rangle = |-+\rangle,\ |4\rangle = |--\rangle$

then

$$\mathcal{H}_\alpha = \begin{bmatrix} \frac{C}{2} - \frac{\omega_n}{2} + \frac{A}{4} & \frac{B}{4} \\ \frac{B^*}{4} & \frac{C}{2} + \frac{\omega_n}{2} - \frac{A}{4} \end{bmatrix} \tag{D.4}$$

and

$$\mathcal{H}_\beta = \begin{bmatrix} -\frac{C}{2} - \frac{\omega_n}{2} - \frac{A}{4} & -\frac{B}{4} \\ -\frac{B^*}{4} & -\frac{C}{2} + \frac{\omega_n}{2} + \frac{A}{4} \end{bmatrix} \tag{D.5}$$

where

$$\mathcal{H}_{\alpha,\beta} \equiv \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{D.6}$$

1). Eigenvalues

The eigenvalues of $\mathcal{H}_\alpha$ and $\mathcal{H}_\beta$ are obtained by transformations

$$T_\alpha^\dagger \mathcal{H}_\alpha T_\alpha = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{D.7}$$

and

$$T_\beta^\dagger \mathcal{H}_\beta T_\beta = \begin{bmatrix} \lambda_3 & 0 \\ 0 & \lambda_4 \end{bmatrix} \tag{D.8}$$

Or from

$$\mathcal{H}_{\alpha,\beta} - \lambda = 0 \tag{D.9}$$

we get

$$\begin{bmatrix} a-\lambda & b \\ b & c-\lambda \end{bmatrix} = 0 \tag{D.10}$$

the solution

$$\lambda = \frac{a+c}{2} \pm \sqrt{\left(\frac{a-c}{2}\right)^2 + b^2} \tag{D.11}$$

Define

$$\gamma \equiv \sqrt{\left(\frac{a-c}{2}\right)^2 + b^2} \tag{D.12}$$

and

$$\frac{c-a}{2} \equiv \gamma \cos\alpha \tag{D.13}$$

Or may be

$$\frac{a-c}{2} \equiv \gamma \cos\alpha \tag{D.14}$$

because $a - c > 0$ for malonic acid.

$$b \equiv \gamma \sin\alpha \tag{D.15}$$

and

$$\lambda = \frac{a+c}{2} \pm \gamma \tag{D.16}$$

For $\mathcal{H}_\alpha$

$$c - a = \omega_n - \frac{A}{2} \tag{D.17}$$

and let

$$\gamma_\alpha \equiv \sqrt{\left(\frac{a-c}{2}\right)^2 + b^2} = \sqrt{\frac{1}{4}\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{4}\right)^2} = \frac{1}{2}\sqrt{\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2} \tag{D.18}$$

$$\omega_\alpha \equiv 2\gamma_\alpha = \sqrt{\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2} \tag{D.19}$$

we get

$$\cos\alpha = \frac{c-a}{2\gamma} = \frac{\omega_n - \frac{A}{2}}{\sqrt{\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2}} \tag{D.20}$$

$$\sin\alpha = \frac{b}{\gamma} = \frac{B}{4\sqrt{\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2}} \tag{D.21}$$

For $\mathcal{H}_\beta$

$$c-a=\omega_n+\frac{A}{2} \tag{D.22}$$

define

$$\gamma_\beta \equiv \sqrt{\left(\frac{a-c}{2}\right)^2+b^2}=\sqrt{\frac{1}{4}\left(\omega_n+\frac{A}{2}\right)^2+\left(\frac{B}{4}\right)^2}=\frac{1}{2}\sqrt{\left(\omega_n+\frac{A}{2}\right)^2+\left(\frac{B}{2}\right)^2} \tag{D.23}$$

$$\omega_\beta \equiv 2\gamma_\beta=\sqrt{\left(\omega_n+\frac{A}{2}\right)^2+\left(\frac{B}{2}\right)^2} \tag{D.24}$$

get

$$\cos\beta=\frac{c-a}{2\gamma}=\frac{\omega_n+\frac{A}{2}}{\sqrt{\left(\omega_n+\frac{A}{2}\right)^2+\left(\frac{B}{2}\right)^2}} \tag{D.25}$$

$$\sin\beta=\frac{b}{\gamma}=-\frac{B}{4\sqrt{\left(\omega_n+\frac{A}{2}\right)^2+\left(\frac{B}{2}\right)^2}} \tag{D.26}$$

from eigenstates

$$E=\begin{array}{c|cccc} & |1\rangle=|a\rangle & |2\rangle=|b\rangle & |3\rangle=|c\rangle & |4\rangle=|d\rangle \\ \hline |1\rangle=|a\rangle & \frac{C}{2}+\frac{1}{2}\omega_\alpha & 0 & 0 & 0 \\ |2\rangle=|b\rangle & 0 & \frac{C}{2}-\frac{1}{2}\omega_\alpha & 0 & 0 \\ |3\rangle=|c\rangle & 0 & 0 & -\frac{C}{2}+\frac{1}{2}\omega_\beta & 0 \\ |4\rangle=|d\rangle & 0 & 0 & 0 & -\frac{C}{2}-\frac{1}{2}\omega_\beta \end{array} \tag{D.27}$$

Define :

$$\begin{aligned} E_1&=E_a=\frac{C}{2}+\frac{1}{2}\omega_\alpha \\ E_2&=E_b=\frac{C}{2}-\frac{1}{2}\omega_\alpha \\ E_3&=E_c=-\frac{C}{2}+\frac{1}{2}\omega_\beta \\ E_4&=E_d=-\frac{C}{2}-\frac{1}{2}\omega_\beta \end{aligned} \tag{D.28}$$

$$\omega_{ij} = E_i - E_j \tag{D.29}$$

$$\omega_{ij} = -\omega_{ji} \tag{D.30}$$

$$\omega_{ij} + \omega_{jk} = \omega_{ik} \tag{D.31}$$

$$E = \begin{array}{r} \\ |1\rangle = |a\rangle \\ |2\rangle = |b\rangle \\ |3\rangle = |c\rangle \\ |4\rangle = |d\rangle \end{array} \begin{array}{c} \begin{array}{cccc} |1\rangle = |a\rangle & |2\rangle = |b\rangle & |3\rangle = |c\rangle & |4\rangle = |d\rangle \end{array} \\ \begin{bmatrix} E_1 & 0 & 0 & 0 \\ 0 & E_2 & 0 & 0 \\ 0 & 0 & E_3 & 0 \\ 0 & 0 & 0 & E_4 \end{bmatrix} \end{array} \tag{D.32}$$

2). Eigenvector

$$\begin{bmatrix} a-\lambda & b \\ b & c-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{D.33}$$

then

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^{\dagger} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1^2 + x_2^2 = 1 \tag{D.34}$$

For $\mathcal{H}_\alpha$

$$\begin{aligned} x_2/x_1 &= \frac{\lambda - a}{b} = \frac{1}{b}\left(\frac{a+c}{2} \pm \gamma - a\right) = \frac{1}{b}\left(\frac{c-a}{2} \pm \gamma\right) \\ &= \frac{1}{\gamma \sin\alpha}(\gamma\cos\alpha \pm \gamma) == \frac{1}{\sin\alpha}(\cos\alpha \pm 1) \end{aligned} \tag{D.35}$$

$$\begin{aligned} x_1^2 + x_2^2 &= x_1^2\left[1 + \frac{1}{(\sin\alpha)^2}(\cos\alpha \pm 1)^2\right] \\ &= x_1^2\left[\frac{(\sin\alpha)^2 + (\cos\alpha)^2 \pm 2\cos\alpha + 1}{(\sin\alpha)^2}\right] = 2x_1^2\left[\frac{1 \pm \cos\alpha}{(\sin\alpha)^2}\right] \end{aligned} \tag{D.36}$$

By definition

$$x_1^2 + x_2^2 = 1 \tag{D.37}$$

we get

$$x_1^2 = \frac{(\sin\alpha)^2}{2(1 \pm \cos\alpha)} = \frac{(1+\cos\alpha)(1-\cos\alpha)}{2(1 \pm \cos\alpha)} = \frac{(1 \mp \cos\alpha)}{2} \tag{D.38}$$

$$x_2^2 = 1 - x_1^2 = \frac{(1 \pm \cos\alpha)}{2} \tag{D.39}$$

$$x_1 = \pm\frac{1}{\sqrt{2}}\sqrt{(1\mp\cos\alpha)} = \pm\frac{1}{\sqrt{2}}\left(1\mp\frac{\omega_n-\frac{A}{2}}{\sqrt{\left(\omega_n-\frac{A}{2}\right)^2+\left(\frac{B}{2}\right)^2}}\right)^{\frac{1}{2}}$$
$$= \pm\frac{1}{\sqrt{2}}\left(1\pm\frac{A/2-\omega_n}{\omega_\alpha}\right)^{\frac{1}{2}} \tag{D.40}$$

$$x_2 = \pm\frac{1}{\sqrt{2}}\sqrt{(1\pm\cos\alpha)} = \pm\frac{1}{\sqrt{2}}\left(1\mp\frac{A/2-\omega_n}{\omega_\alpha}\right)^{\frac{1}{2}} \tag{D.41}$$

$$T_\alpha = \begin{array}{c|cc} & |1\rangle=|a\rangle & |2\rangle=|b\rangle \\ |1\rangle=|a\rangle & \multicolumn{2}{c}{\begin{bmatrix} x_1^{(1)} & x_1^{(2)} \\ x_2^{(1)} & x_2^{(2)} \end{bmatrix}} \\ |2\rangle=|b\rangle & & \end{array} \tag{D.42}$$

$$T_\alpha T_\alpha^\dagger = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} \\ x_2^{(1)} & x_2^{(2)} \end{bmatrix}\begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \end{bmatrix}$$
$$= \begin{bmatrix} x_1^{(1)2}+x_1^{(2)2} & x_1^{(1)}x_2^{(1)}+x_1^{(2)}x_2^{(2)} \\ x_2^{(1)}x_1^{(1)}+x_2^{(2)}x_1^{(2)} & x_2^{(1)2}+x_2^{(2)2} \end{bmatrix} \tag{D.43}$$

Choose

$$T_\alpha = \begin{bmatrix} \frac{1}{\sqrt{2}}\sqrt{1+\cos\alpha} & -\frac{1}{\sqrt{2}}\sqrt{1-\cos\alpha} \\ \frac{1}{\sqrt{2}}\sqrt{1-\cos\alpha} & \frac{1}{\sqrt{2}}\sqrt{1+\cos\alpha} \end{bmatrix} \tag{D.44}$$

$$T_\alpha T_\alpha^\dagger = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} \\ x_2^{(1)} & x_2^{(2)} \end{bmatrix}\begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \end{bmatrix}$$
$$= \begin{bmatrix} x_1^{(1)2}+x_1^{(2)2} & x_1^{(1)}x_2^{(1)}+x_1^{(2)}x_2^{(2)} \\ x_2^{(1)}x_1^{(1)}+x_2^{(2)}x_1^{(2)} & x_2^{(1)2}+x_2^{(2)2} \end{bmatrix} \tag{D.45}$$

$$= \begin{bmatrix} \frac{1+\cos\alpha}{2}+\frac{1-\cos\alpha}{2} & \frac{\sqrt{1+\cos\alpha}\sqrt{1-\cos\alpha}}{2}-\frac{\sqrt{1-\cos\alpha}\sqrt{1+\cos\alpha}}{2} \\ \frac{\sqrt{1-\cos\alpha}\sqrt{1+\cos\alpha}}{2}-\frac{\sqrt{1+\cos\alpha}\sqrt{1-\cos\alpha}}{2} & \frac{1-\cos\alpha}{2}+\frac{1+\cos\alpha}{2} \end{bmatrix} \tag{D.46}$$

$$T_\alpha T_\alpha^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{D.47}$$

Similarity for $\mathcal{H}_\beta$

$$x_3 = \pm\frac{1}{\sqrt{2}}\sqrt{(1 \mp \cos\beta)} = \pm\frac{1}{\sqrt{2}}\left(1 \mp \frac{\omega_n + \frac{A}{2}}{\sqrt{\left(\omega_n + \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2}}\right)^{\frac{1}{2}} \tag{D.48}$$

$$= \pm\frac{1}{\sqrt{2}}\left(1 \mp \frac{A/2 + \omega_n}{\omega_\beta}\right)^{\frac{1}{2}}$$

$$x_4 = \pm\frac{1}{\sqrt{2}}\sqrt{(1 \pm \cos\beta)} = \pm\frac{1}{\sqrt{2}}\left(1 \pm \frac{A/2 + \omega_n}{\omega_\beta}\right)^{\frac{1}{2}} \tag{D.49}$$

$$T_\beta = \begin{array}{cc} & \begin{array}{cc} |3\rangle = |c\rangle & |4\rangle = |d\rangle \end{array} \\ \begin{array}{c} |3\rangle = |c\rangle \\ |4\rangle = |d\rangle \end{array} & \begin{bmatrix} x_3^{(1)} & x_3^{(2)} \\ x_4^{(1)} & x_4^{(2)} \end{bmatrix} \end{array} \tag{D.50}$$

Choose

$$T_\beta = \begin{bmatrix} \frac{1}{\sqrt{2}}\sqrt{1+\cos\beta} & \frac{1}{\sqrt{2}}\sqrt{1-\cos\beta} \\ -\frac{1}{\sqrt{2}}\sqrt{1-\cos\beta} & \frac{1}{\sqrt{2}}\sqrt{1+\cos\beta} \end{bmatrix} \tag{D.51}$$

$$\cos\alpha = \frac{\omega_n - \frac{A}{2}}{\sqrt{\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2}} \tag{D.52}$$

$$\cos\beta = \frac{\omega_n + \frac{A}{2}}{\sqrt{\left(\omega_n + \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2}} \tag{D.53}$$

$$T_\beta T_\beta^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{D.54}$$

$$P = \begin{array}{cccc} |1\rangle = |a\rangle & |2\rangle = |b\rangle & |3\rangle = |c\rangle & |4\rangle = |d\rangle \\ \left[\begin{matrix} \frac{1}{\sqrt{2}}\sqrt{1+\cos\alpha} & -\frac{1}{\sqrt{2}}\sqrt{1-\cos\alpha} & 0 & 0 \\ \frac{1}{\sqrt{2}}\sqrt{1-\cos\alpha} & \frac{1}{\sqrt{2}}\sqrt{1+\cos\alpha} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}}\sqrt{1+\cos\beta} & \frac{1}{\sqrt{2}}\sqrt{1-\cos\beta} \\ 0 & 0 & -\frac{1}{\sqrt{2}}\sqrt{1-\cos\beta} & \frac{1}{\sqrt{2}}\sqrt{1+\cos\beta} \end{matrix}\right. & & & \!\!\!\!\!\!\!\!\left.\right] \end{array} \tag{D.55}$$

$$P = \begin{array}{c} (\ |a\rangle, |b)(\ |c\rangle, |d) \\ \begin{bmatrix} T_\alpha & \mathbf{0} \\ \mathbf{0} & T_\beta \end{bmatrix} \end{array} \tag{D.56}$$

where

$$\mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{D.57}$$

$$PP^\dagger = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \tag{D.58}$$

$$\boldsymbol{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{D.59}$$

3). Matrix $M = T_\alpha^\dagger T_\beta$

$$\begin{aligned} M = T_\alpha^\dagger T_\beta &= \begin{bmatrix} \frac{1}{\sqrt{2}}\sqrt{1+\cos\alpha} & \frac{1}{\sqrt{2}}\sqrt{1-\cos\alpha} \\ -\frac{1}{\sqrt{2}}\sqrt{1-\cos\alpha} & \frac{1}{\sqrt{2}}\sqrt{1+\cos\alpha} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}}\sqrt{1+\cos\beta} & \frac{1}{\sqrt{2}}\sqrt{1-\cos\beta} \\ -\frac{1}{\sqrt{2}}\sqrt{1-\cos\beta} & \frac{1}{\sqrt{2}}\sqrt{1+\cos\beta} \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} \sqrt{1+\cos\alpha} & \sqrt{1-\cos\alpha} \\ -\sqrt{1-\cos\alpha} & \sqrt{1+\cos\alpha} \end{bmatrix} \begin{bmatrix} \sqrt{1+\cos\beta} & \sqrt{1-\cos\beta} \\ -\sqrt{1-\cos\beta} & \sqrt{1+\cos\beta} \end{bmatrix} \end{aligned} \tag{D.60}$$

$$M_{11} = \frac{1}{2}\left[\sqrt{(1+\cos\alpha)(1+\cos\beta)} - \sqrt{(1-\cos\alpha)(1-\cos\beta)}\right] \tag{D.61}$$

$$\begin{aligned}4M_{11}^2 &= (1+\cos\alpha)(1+\cos\beta)+(1-\cos\alpha)(1-\cos\beta)\\ &\qquad -2\sqrt{(1+\cos\alpha)(1+\cos\beta)(1-\cos\alpha)(1-\cos\beta)}\\ &= 2+2\cos\alpha\cos\beta-2\sqrt{(\sin\alpha\sin\beta)^2}\\ &= 2+2\cos\alpha\cos\beta-2|\sin\alpha\sin\beta|\end{aligned} \tag{D.62}$$

because

$$\sin\alpha\sin\beta = \frac{B}{4\sqrt{\left(\omega_n-\frac{A}{2}\right)^2+\left(\frac{B}{2}\right)^2}}\frac{-B}{4\sqrt{\left(\omega_n+\frac{A}{2}\right)^2+\left(\frac{B}{2}\right)^2}} < 0 \tag{D.63}$$

thus

$$|\sin\alpha\sin\beta| = -\sin\alpha\sin\beta \tag{D.64}$$

$$2M_{11}^2 = 1+\cos\alpha\cos\beta+\sin\alpha\sin\beta = 1+\cos(\alpha-\beta) \tag{D.65}$$

$$M_1^2 \equiv M_{11}^2 = \frac{1}{2}[1+\cos(\alpha-\beta)] \tag{D.66}$$

$$M_{12} = \frac{1}{2}\left[\sqrt{(1+\cos\alpha)(1-\cos\beta)}+\sqrt{(1-\cos\alpha)(1+\cos\beta)}\right] \tag{D.67}$$

$$\begin{aligned}4M_{12}^2 &= (1+\cos\alpha)(1-\cos\beta)+(1-\cos\alpha)(1+\cos\beta)\\ &\qquad +2\sqrt{(1+\cos\alpha)(1-\cos\beta)(1-\cos\alpha)(1+\cos\beta)}\\ &\qquad = 2-2\cos\alpha\cos\beta+2\sqrt{(\sin\alpha\sin\beta)^2}\\ &\qquad = 2-2\cos\alpha\cos\beta-2\sin\alpha\sin\beta = 2(1-\cos(\alpha-\beta))\end{aligned} \tag{D.68}$$

$$M_2^2 \equiv M_{12}^2 = \frac{1}{2}[1-\cos(\alpha-\beta)] \tag{D.69}$$

$$M_1^2M_2^2 = \frac{1}{4}[1+\cos(\alpha-\beta)][1-\cos(\alpha-\beta)] = \frac{\sin(\alpha-\beta)^2}{4} \tag{D.70}$$

$$\begin{aligned}&\sin(\alpha-\beta) = \sin\alpha\cos\beta-\cos\alpha\sin\beta\\ &= \left(\frac{b}{\gamma}\right)_\alpha\left(\frac{c-a}{2\gamma}\right)_\beta-\left(\frac{b}{\gamma}\right)_\beta\left(\frac{c-a}{2\gamma}\right)_\alpha\\ &= \frac{1}{2\gamma_\alpha\gamma_\beta}\left[b_\alpha(c_\beta-a_\beta)-b_\beta(c_\alpha-a_\alpha)\right] = \frac{1}{2\gamma_\alpha\gamma_\beta}\left[\frac{B}{4}\left(\omega_n-\frac{A}{2}\right)-\frac{-B}{4}\left(\omega_n+\frac{A}{2}\right)\right]\\ &\qquad\qquad = \frac{B\omega_n}{4\gamma_\alpha\gamma_\beta}\\ &= \frac{B\omega_n}{\omega_\alpha\omega_\beta}\end{aligned} \tag{D.71}$$

$$M_1^2 M_2^2 = \frac{1}{4}\left(\frac{B\omega_n}{\omega_\alpha \omega_\beta}\right)^2 \tag{D.72}$$

If $\omega_n = \frac{A}{2}$ then

$$\sin(\alpha - \beta) = \frac{AB}{2\omega_\alpha \omega_\beta} \tag{D.73}$$

$$|\cos(\alpha - \beta)| = \sqrt{1 - (\sin(\alpha - \beta))^2} = \sqrt{1 - \left(\frac{B\omega_n}{\omega_\alpha \omega_\beta}\right)^2} \tag{D.74}$$

or

$$\begin{aligned}
&\cos(\alpha - \beta) = \cos\alpha \cos\beta + \sin\alpha \sin\beta \\
&= \left(\frac{c-a}{2\gamma}\right)_\alpha \left(\frac{c-a}{2\gamma}\right)_\beta + \left(\frac{b}{\gamma}\right)_\alpha \left(\frac{b}{\gamma}\right)_\beta \\
&= \frac{1}{\gamma_\alpha \gamma_\beta}\left[\frac{1}{2}\left(\omega_n - \frac{A}{2}\right)\frac{1}{2}\left(\omega_n + \frac{A}{2}\right) + \left(\frac{B}{4}\right)\left(-\frac{B}{4}\right)\right] \\
&= \frac{1}{4\gamma_\alpha \gamma_\beta}\left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right] \\
&= \frac{1}{\omega_\alpha \omega_\beta}\left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]
\end{aligned} \tag{D.75}$$

$$M_1^2 = \frac{1}{2}[1 + \cos(\alpha - \beta)] = \frac{1}{2}\left[1 + \frac{\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2}{\omega_\alpha \omega_\beta}\right] \tag{D.76}$$

$$M_2^2 = \frac{1}{2}[1 - \cos(\alpha - \beta)] = \frac{1}{2}\left[1 - \frac{\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2}{\omega_\alpha \omega_\beta}\right] \tag{D.77}$$

Or for a symmetry of calculation as in Gamliel and Freed (1990) paper

$$M_1^2 = \frac{1}{2}\left[1 \mp \frac{\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2}{\omega_\alpha \omega_\beta}\right] \tag{D.78}$$

$$M_2^2 = \frac{1}{2}\left[1 \pm \frac{\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2}{\omega_\alpha \omega_\beta}\right] \tag{D.79}$$

$$\begin{aligned}
[1+\cos(\alpha-\beta)]^2 &= \left[1 + \frac{1}{\omega_\alpha\omega_\beta}\left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \\
&= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\omega_\alpha\omega_\beta + \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2
\end{aligned} \tag{D.80}$$

$$\begin{aligned}
&= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left[\left(\omega_\alpha + \omega_\beta\right)^2 - \omega_\alpha^2 - \omega_\beta^2\right] + \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \\
&= \frac{1}{\omega_\alpha^2\omega_\beta^2}\Bigg[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - \frac{1}{2}\left[\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2 + \left(\omega_n + \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2\right] \\
&\qquad + \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\Bigg]^2
\end{aligned} \tag{D.81}$$

$$= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - \frac{1}{2}\left[2\omega_n^2 + 2\left(\frac{A}{2}\right)^2 + 2\left(\frac{B}{2}\right)^2\right] + \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \tag{D.82}$$

$$\begin{aligned}
&= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - \left[\omega_n^2 + \left(\frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2\right] + \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \\
&= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 + \left[-\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right] + \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2
\end{aligned} \tag{D.83}$$

$$\begin{aligned}
&= \frac{1}{\omega_\alpha^2\omega_\beta^2}\Bigg[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 + \left[-\omega_n^2 + \omega_n A - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right] + 2\omega_n^2 \\
&\qquad + \left[-\omega_n^2 - \omega_n A - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\Bigg]^2
\end{aligned} \tag{D.84}$$

$$= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - \left[\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2\right] + 2\omega_n^2 - \left[\left(\omega_n + \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2\right]\right]^2 \tag{D.85}$$

$$\begin{aligned} &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 + 2\omega_n^2 - \omega_\alpha^2 - \omega_\beta^2\right]^2 \\ &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[2\omega_n^2 - \frac{1}{2}\left(\omega_\alpha - \omega_\beta\right)^2\right]^2 \end{aligned} \tag{D.86}$$

$$\begin{aligned} M_1^4 = \frac{1}{4}[1 + \cos(\alpha - \beta)]^2 &= \frac{1}{4}\frac{1}{\omega_\alpha^2\omega_\beta^2}\left[2\omega_n^2 - \frac{1}{2}\left(\omega_\alpha - \omega_\beta\right)^2\right]^2 \\ &= \left[\frac{\omega_n^2 - \frac{1}{4}\left(\omega_\alpha - \omega_\beta\right)^2}{\omega_\alpha\omega_\beta}\right]^2 \end{aligned} \tag{D.87}$$

$$\begin{aligned} [1 - \cos(\alpha - \beta)]^2 &= \left[1 - \frac{1}{\omega_\alpha\omega_\beta}\left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \\ &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\omega_\alpha\omega_\beta - \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \end{aligned} \tag{D.88}$$

$$\begin{aligned} &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left[\left(\omega_\alpha + \omega_\beta\right)^2 - \omega_\alpha^2 - \omega_\beta^2\right] - \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \\ &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - \frac{1}{2}\left[\left(\omega_n - \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2 + \left(\omega_n + \frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2\right]\right. \\ &\qquad \left. - \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \end{aligned} \tag{D.89}$$

$$\begin{aligned} &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - \frac{1}{2}\left[2\omega_n^2 + 2\left(\frac{A}{2}\right)^2 + 2\left(\frac{B}{2}\right)^2\right] - \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \\ &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - \left[\omega_n^2 + \left(\frac{A}{2}\right)^2 + \left(\frac{B}{2}\right)^2\right] - \left[\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2\right]\right]^2 \\ &= \frac{1}{\omega_\alpha^2\omega_\beta^2}\left[\frac{1}{2}\left(\omega_\alpha + \omega_\beta\right)^2 - 2\omega_n^2\right]^2 \end{aligned} \tag{D.90}$$

$$M_2^4 = \frac{1}{4}[1-\cos(\alpha-\beta)]^2 = \frac{1}{4}\frac{1}{\omega_\alpha^2\omega_\beta^2}\left[2\omega_n^2 - \frac{1}{2}\left(\omega_\alpha+\omega_\beta\right)^2\right]^2$$
$$= \left[\frac{\omega_n^2 - \frac{1}{4}\left(\omega_\alpha+\omega_\beta\right)^2}{\omega_\alpha\omega_\beta}\right]^2 \quad \text{(D.91)}$$

Define

$$k \equiv 4M_1^2M_2^2 = \left(\frac{B\omega_n}{\omega_\alpha\omega_\beta}\right)^2 \quad \text{(D.92)}$$

Let

$$u = \cos(\alpha-\beta) \quad \text{(D.93)}$$

$$M_1^2 = \frac{1}{2}(1+u) \quad \text{(D.94)}$$

$$M_2^2 = \frac{1}{2}(1-u) \quad \text{(D.95)}$$

$$\frac{1}{4}(1+u)(1-u) = \frac{1}{4}k \quad \text{(D.96)}$$

$$u^2 = 1-k \quad \text{(D.97)}$$

$$u = \sqrt{1-k} \quad \text{(D.98)}$$

$$M_1^4 = \frac{1}{4}[1+\cos(\alpha-\beta)]^2 = \frac{1}{4}\left[1+\sqrt{1-k}\right]^2 = \frac{1}{2}\left(1-\frac{k}{2}+\sqrt{1-k}\right)$$
$$\equiv k_+ \quad \text{(D.99)}$$

$$M_2^4 = \frac{1}{4}[1-\cos(\alpha-\beta)]^2 = \frac{1}{4}\left[1-\sqrt{1-k}\right]^2$$
$$= \frac{1}{2}\left(1-\frac{k}{2}-\sqrt{1-k}\right) \equiv k_- \quad \text{(D.100)}$$

$$M_1^2 = \frac{1}{2}[1+\cos(\alpha-\beta)] = \frac{1}{2}\left(1+\sqrt{1-k}\right) \quad \text{(D.101)}$$

$$M_2^2 = \frac{1}{2}[1 - \cos(\alpha - \beta)] = \frac{1}{2}\left(1 - \sqrt{1-k}\right) \tag{D.102}$$

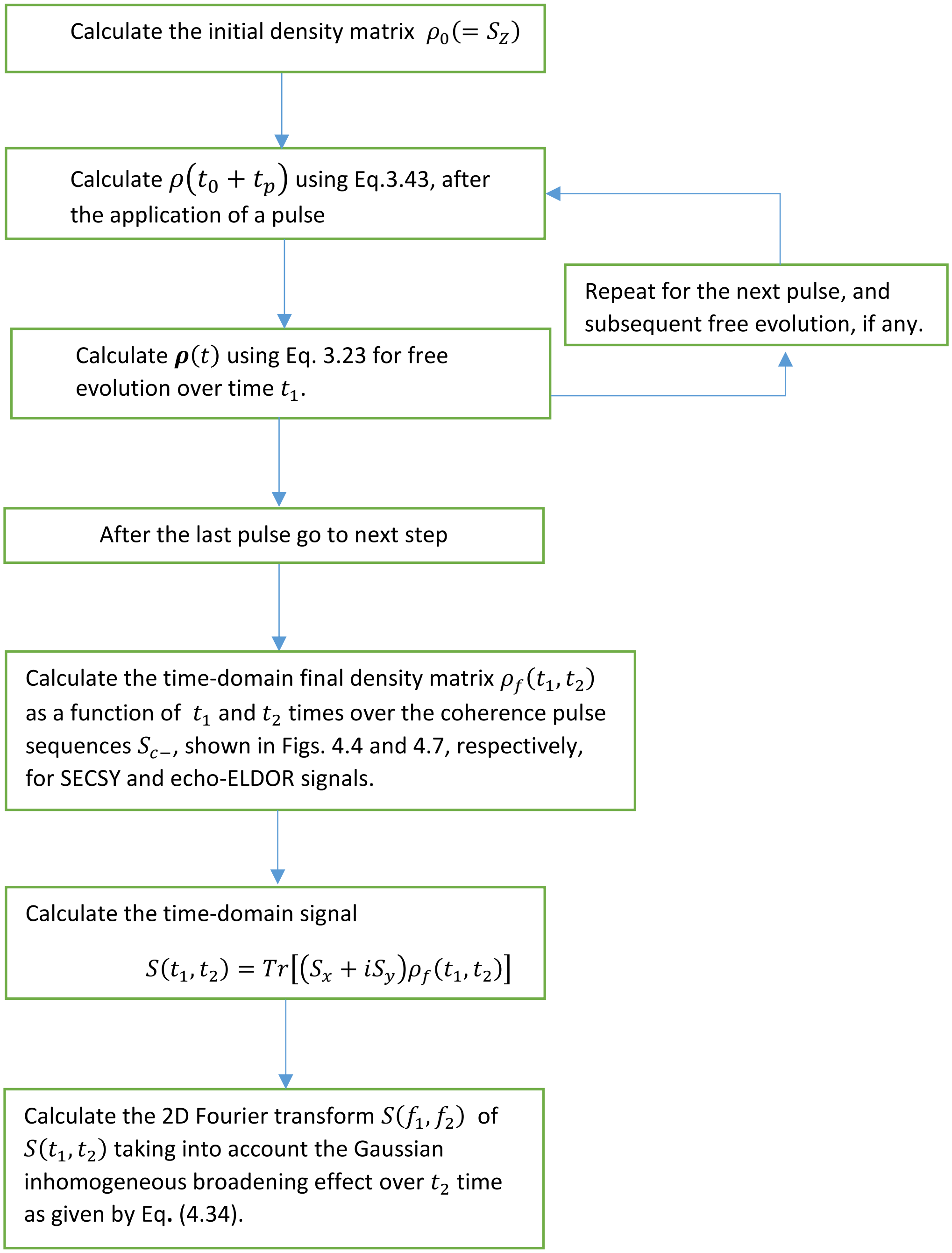
$$M = T_\alpha^\dagger T_\beta = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \tag*{\textbf{(D.103)}}$$

$$M \equiv \begin{bmatrix} m_1 & m_2 \\ -m_2 & m_1 \end{bmatrix} \tag{D.104}$$

$$m_1 = \cos\varphi = \left(\frac{1 + \cos(\alpha - \beta)}{2}\right)^{\frac{1}{2}} = \left(\frac{\left(1 + \sqrt{1-k}\right)}{2}\right)^{\frac{1}{2}} \tag{D.105}$$

$$m_2 = \sin\varphi = \left(\frac{1 - \cos(\alpha - \beta)}{2}\right)^{\frac{1}{2}} = \left(\frac{\left(1 - \sqrt{1-k}\right)}{2}\right)^{\frac{1}{2}} \tag{D.106}$$

Appendix E. Flow chart for computing SECSY and echo-ELDOR signals
The Figure below describes the various steps in the simulation.


Calculate the initial density matrix $\rho_0(= S_Z)$
Calculate $\rho\left(t_0 + t_p\right)$ using Eq.3.43, after the application of a pulse
Repeat for the next pulse, and subsequent free evolution, if any.
Calculate $\boldsymbol{\rho}(t)$ using Eq. 3.23 for free evolution over time t_1.
After the last pulse go to next step
Calculate the time-domain final density matrix $\rho_f(t_1, t_2)$ as a function of t_1 and t_2 times over the coherence pulse sequences S_{c-}, shown in Figs. 4.4 and 4.7, respectively, for SECSY and echo-ELDOR signals.
Calculate the time-domain signal $S(t_1, t_2) = Tr\left[\left(S_x + iS_y\right)\rho_f(t_1, t_2)\right]$
Calculate the 2D Fourier transform $S(f_1, f_2)$ of $S(t_1, t_2)$ taking into account the Gaussian inhomogeneous broadening effect over t_2 time as given by Eq. (4.34).

Appendix F. The various terms in the spin-fluctuation Hamiltonian H_1 as given by Eq. (4.1) [Lee *et al.* (1993)]

Here the subscripts a, b, c and d correspond to the eigenvalues E_a, E_b, E_c and E_d for the spin Hamiltonian as given by Eq. (D.28). For the notation for the square of matrix elements e.g., $|A_{ac}|^2 = |\langle a|A|c\rangle|^2$. The fluctuations of SHP $\left(\delta g, \delta a, \delta F, \delta F^{(2)}, \delta D, \delta F^{(2)}\right)$ are the time-dependent deviations from the average values, e.g. $\delta g = g(t) - \bar{g}$. The relevant coefficients for the nuclear modulation (row 3 of the table) are: $m_1^2 = \frac{1}{2}\left[1 + \frac{\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2}{\omega_\alpha \omega_\beta}\right]$, $m_2^2 = \frac{1}{2}\left[1 - \frac{\omega_n^2 - \left(\frac{A}{2}\right)^2 - \left(\frac{B}{2}\right)^2}{\omega_\alpha \omega_\beta}\right]$

	F^q	A^q	$\lvert A_{ac}\rvert^2$	$\lvert A_{bd}\rvert^2$	$\lvert A_{ad}\rvert^2$	$\lvert A_{bc}\rvert^2$	$\lvert A_{ab}\rvert^2$	$\lvert A_{cd}\rvert^2$	$\lvert A_{aa}\rvert^2$, $\lvert A_{bb}\rvert^2$	$\lvert A_{cc}\rvert^2$, $\lvert A_{dd}\rvert^2$
Isotropic g	$\frac{\beta_e\, B_0}{\hbar}\delta g$	S_z							$\frac{1}{4}$	$\frac{1}{4}$
Anisotropic g	$\delta F\frac{1}{2}(3\cos^2\beta - 1)$ $+ \delta F^{(2)}\sin^2\beta\cos(2\gamma)$	S_z							$\frac{1}{4}$	$\frac{1}{4}$
	$\delta F\frac{3}{8}\sin(2\beta)$ $- \delta F^{(2)}\frac{1}{4}\sin(2\beta)\cos(2$	S_+ $+ S_-$	m_1^2	m_1^2	m_1^2	m_1^2				
Isotropic hf	$-\frac{\gamma_e}{\hbar}\delta a$	S_zI_z					$\frac{1}{4}[Re(c_1^*c_2)]$	$\frac{1}{4}[Re(c_3^*c_4)]$	$\frac{1}{16}(\lvert c_1\rvert^2$ $- \lvert c_2\rvert^2)^2$	$\frac{1}{16}(\lvert c_3\rvert^2$ $- \lvert c_4\rvert^2)^2$
		$\frac{1}{2}(S_+I_-$ $+ S_-I_+)$	$\frac{1}{4}\lvert c_2^*c_3\rvert^2$	$\frac{1}{4}\lvert c_1^*c_4\rvert^2$	$\frac{1}{4}\lvert c_2^*c_4\rvert^2$	$\frac{1}{4}\lvert c_1^*c_3\rvert^2$				
Anisotropic hf	$-\frac{\gamma_e}{\hbar}\Big[\delta D\frac{1}{2}(3\cos^2\beta$ $- 1)$ $+ \delta D^{(2)}\sin^2\beta\cos(2\gamma)\Big]$	S_zI_z					$\frac{1}{4}[Re(c_1^*c_2)]$	$\frac{1}{4}[Re(c_3^*c_4)]$	$\frac{1}{16}(\lvert c_1\rvert^2$ $- \lvert c_2\rvert^2)^2$	$\frac{1}{16}(\lvert c_3\rvert^2$ $- \lvert c_4\rvert^2)^2$
		$-\frac{1}{4}(S_+I_-$ $+ S_-I_+)$	$\frac{1}{16}\lvert c_2^*c_3\rvert^2$	$\frac{1}{16}\lvert c_1^*c_4\rvert^2$	$\frac{1}{16}\lvert c_2^*c_4\rvert^2$	$\frac{1}{16}\lvert c_1^*c_3\rvert^2$				
	$-\frac{\gamma_e}{\hbar}\Big[\delta D\frac{3}{8}\sin(2\beta)$ $- \delta D^{(2)}\frac{1}{4}\sin(2\beta)\cos(2$	S_zI_+ $+ S_zI_-$					$\frac{1}{4}(\lvert c_1\rvert^2$ $- \lvert c_2\rvert^2)^2$	$\frac{1}{4}(\lvert c_3\rvert^2$ $- \lvert c_4\rvert^2)^2$	$[Re(c_1^*c_2)]^2$	$[Re(c_3^*c_4)]^2$
		S_+I_z $+ S_-I_z$	$\frac{1}{4}\lvert c_1^*c_3$ $- c_2^*c_4\rvert^2$	$\frac{1}{4}\lvert c_1^*c_3$ $- c_2^*c_4\rvert^2$	$\frac{1}{4}\lvert c_1^*c_4$ $+ c_2^*c_3\rvert^2$	$\frac{1}{4}\lvert c_1^*c_4$ $+ c_2^*c_3\rvert^2$				
	$-\frac{\gamma_e}{\hbar}\Big[\delta D\frac{3}{8}\sin^2$ $- \delta D^{(2)}\frac{1}{4}(1$ $+ \cos^2\beta)\cos(2\gamma)\Big]$	S_+I_+ $+ S_-I_-$	$\lvert c_1^*c_4\rvert^2$	$\lvert c_2^*c_3\rvert^2$	$\lvert c_1^*c_3\rvert^2$	$\lvert c_2^*c_4\rvert^2$				

Appendix G. Matlab source code

1. main function

```
fclose('all'); % Close all open files
clear % Remove items from MATLAB workspace and reset MuPAD engine
close all hidden % Deletes all Figures including those with hidden handles.
clc %  Clears all input and output from the Command Window display


%Sample Radiated Malonic Aci [McConnell, Heller,Cole and Fessenden.
%Journal of the American Chemical Society 82, no. 4 (1960): 766-775.]
B0 = 0; %Rotating frame; 3000.0d0 Gauss, Gambliel 1990
Wn = 2*pi*14.5; % w_n the nuclear Larmor frequency for the nucleus uint: MHz(*rad)
g_tensor = [2.0026d0, 2.0035d0, 2.0033d0];%[2.0026d0, 2.0035d0, 2.0033d0];
hyperfine_tensor = [-61.0d0, -91.0d0, -29.0d0]; % [-61.0d0, -91.0d0, -29.0d0] MHz for ZX plan,
Fig.8 of [2]
T2e = 0.687d0;  % T2 Spin–spin relaxation, uint: ms

orientation = 2;
if orientation > 3
   hyperfine_tensor = [-91.0d0, -61.0d0, -29.0d0]; %MHz for ZY plan, exchange Ax and Ay for
ZY plan, Fig.9 of [2]
end
switch orientation
   % Following for ZX plan, Fig.8 of [2]
   case 1 % Fig.8a and Table 1 of [2]
      euler_angles = [0.0d0,  -5.0d0 * pi/180.0d0, 0.0d0 * pi/180.0d0 ]; %rad, Internal angle
(theta=05, ph=0) in zx quadrant.
      T2e = 0.800d0;  % T2 Spin–spin relaxation, unit: microsecond
      orientationTitle =  ['Orientation(5' char(176) ', 0' char(176) ')'];
   case 2 % Fig.8b and Table 1 of [2]
      euler_angles = [0.0d0, -30.0d0 * pi/180.0d0, 0.0d0 * pi/180.0d0 ]; %rad, Internal angle
(theta=30, ph=0) in zx quadrant.
      T2e = 0.812d0;  % T2 Spin–spin relaxation, unit: microsecond
      orientationTitle =  ['Orientation(30' char(176) ', 0' char(176) ')'];
   case 3 % Fig.8c and Table 1 of [2]
      euler_angles = [0.0d0, -50.0d0 * pi/180.0d0, 0.0d0 * pi/180.0d0 ]; %rad, Internal angle
(theta=50, ph=0) in zx quadrant.
      T2e = 0.687d0;  % T2 Spin–spin relaxation, unit: microsecond
      orientationTitle =  ['Orientation(50' char(176) ', 0' char(176) ')'];

      % Following for ZY plan, Fig.9 of [2]
   case 4 % Fig.9a of and Table 1 of [2]
      euler_angles = [0.0d0,  5.0d0 * pi/180.0d0, 90.0d0 * pi/180.0d0 ]; %rad, Internal angle
(theta=05, ph=90) in zy quadrant.
```

```
        T2e = 1.223d0;  % T2 Spin–spin relaxation, unit: microsecond
        orientationTitle =  ['Orientation(5' char(176) ', 90' char(176) ')'];
    case 5 % Fig.9b and Table 1 of [2]
        euler_angles = [0.0d0, 45.0d0 * pi/180.0d0, 90.0d0 * pi/180.0d0 ]; %rad, Internal angle
(theta=45, ph=90) in zy quadrant.
        T2e = 1.068d0;  % T2 Spin–spin relaxation, unit: microsecond
        orientationTitle =  ['Orientation(45' char(176) ', 90' char(176) ')'];
    otherwise
        error('No such orientation!')
end %switch orientation

disp(['Orientation: ', 'theta= ',  num2str( euler_angles(2)*180/pi ),char(176) , ' phi= ',
num2str( euler_angles(3)*180/pi), char(176)]); % %degree symbol char(176)

hamiltonianFactor = HamiltonianFactor_class(B0, Wn, g_tensor, hyperfine_tensor, euler_angles);
%hamiltonianFactor.Print();

%Lee, Sanghyuk, Baldev R. Patyal, and Jack H. Freed.
%"A two?dimensional Fourier transform electron?spin resonance (ESR) study of nuclear
modulation and spin relaxation in irradiated malonic acid."
%The Journal of chemical physics 98.5 (1993): 3665-3689]
% Data from Fig.11 of [2]

delta = 4.0d0; % 5 Gaussian inhomogeneous broadening effect, unit: MHz

t2e = 0.9d0; %T2e Off-diagonal electron spin–spin relaxation, unit: microsecond
t2n = 22.0d0; % T2n Off-diagonal spin–nuclear relaxation, unit: microsecond
we = 1.67E-02; % Diagonal Lattice induced electron-spin flip ralaxation rates, unit:
1/microsecond
wn = 7.14E-03; % Diagonal Lattice induced nuclear-spin flip ralaxation rates, unit:
1/microsecond
wx = 6.17E-03; % Diagonal Cross relaxation, unit: 1/microsecond
wy = wx; % Diagonal Diagonal Cross relaxation, unit: 1/microsecond
whe = 0.0d0; % Heisenber spin exchange, unit: 1/microsecond

withRelaxation = true;

% LPF A22
relaxation = Relaxation_class(T2e, t2n, we, wn, wx, wy, whe, withRelaxation, withRelaxation);

fluctuation = 1.0E-3; %0.01E01-01=0.1%; the factor of The fluctuating time-dependent
perturbation
tc = 1.0E-1;% correlation time, unit:microsecond, at page 2322 [Freed, 1965]
disp(['Fluctuation = ', num2str(fluctuation),  '; tc = ', num2str(tc)]);
```

```
fluctuationRelaxation = FluctuationRelaxation_class(hamiltonianFactor, T2e, t2n, we, wn, wx,
wy, whe, fluctuation, tc);

relaxationType = 2; % 1: LPF A22; 2: fluctuation relaxation; 3: LPF A22 and fluctuation
relaxation
hamiltonian = Hamiltonian_class(hamiltonianFactor, relaxation, fluctuationRelaxation,
relaxationType);
density = Density_class(hamiltonian);

pulsetime0 = 5.0E-03; % pi/2 pulse( -5 ns, LPF p3667), unit: Microsecond
pulsetype1 = [0.0d0 * pi/180.0d0,  pi/2.0]; % (phase, tipAngle) pair
pulsetype2 = [0.0d0 * pi/180.0d0,  pi]; % (phase, tipAngle) pair

pulse_class = Pulse_class(hamiltonian, density, pulsetime0, pulsetype1, pulsetype2);
fED = FED_class(hamiltonian, density);
pathway_class = Pathway_class(density,hamiltonian);

signalFile = 'signal2dm.txt';
fs1 = 200;                 % Sampling frequency
fs2 = 200;                 % Sampling frequency

plot_class = Plot_class(fs1, fs2);

signalType = 1; % 1 - Sc-, 2 - Sc+
withGaussian = true; % true, Gaussian inhomogeneous broadening effect

signal = Signal_class(hamiltonian, density, pulse_class, fED, relaxation, pathway_class,
signalType, ...
   plot_class.Tv1, plot_class.Tv2, euler_angles, withGaussian, delta, signalFile);

titleName = 'Welcome to 2D Pulses!';
pulse = 3; % 2 or 3 pulses
switch pulse
   case 2
      titleName = 'SECSY ';% orientationTitle];
      pulses2 = {pulsetype1, pulsetype1};
      spinPathway2 = [1 -1];

      signal.Output2Pulses(pulses2, spinPathway2);

   case 3
      titleName = 'ELDOR ';% orientationTitle];

      % Fig.11, Fig.13 and Fig.14 and Table 1 of [2]
      % case orientation = 2;
```

```
        t = 5; %5, 20, 40, 60; %unit: microsecond
        tTitle = [' T = ', num2str(t), 'microsecond'];
        titleName = [titleName, tTitle];

        spinPathway3 =[1 0 -1]; %[1 0 -1];

        pulses3 = {pulsetype1, pulsetype1, pulsetype1};
        signal.Output3Pulses(pulses3, t, spinPathway3);
    otherwise
        error('2 or 3 pulses only!')
end %switch pulse

plot_class.SignalToFFT(signalFile);

rotateDegree = 0;%90 rotateDegree: rotate the plot around z-axis, unit:degree, not radian
plotAll = false; %true, false

plot_class.Plot(titleName, rotateDegree, plotAll);
```

2. Density_class

```
classdef Density_class < handle
    properties
        Hamiltonian;

        EquilibriumH = zeros(4,4); % Equilibrium density operator in Hilbert space in SzIz basis of
H
        EquilibriumL = zeros(16, 1); % Equilibrium density operator in Liouville space in SzIz
basis of H
        DensityL = zeros(16, 1); % Current density operator in Liouville space in SzIz basis of H

    end %properties

    methods
        function this = Density_class(hamiltonian)
            this.Hamiltonian = hamiltonian;
        end
        %Suppose, at initial state, the spin is in in z-axis
        function Init(this)

            equilibriumH = zeros(4,4);

            % Equilibrium density operator in Hilbert space and |SzIz> basis
            e0 = complex(0.50e0, 0.0e0);
```

```
   equilibriumH(1,1) = e0;
   equilibriumH(2,2) = e0;
   equilibriumH(3,3) = -e0;
   equilibriumH(4,4) = -e0;

   % Equilibrium density operator in SzIz basis of H
   this.EquilibriumH = equilibriumH;
   this.EquilibriumL = reshape(this.EquilibriumH, [4*4, 1]);
   this.DensityL =this.EquilibriumL;

end % Density_class constructor

% Liouville space to Hilbert space
function[A] = GetDensityMat(this)

   B = this.DensityL;
   n = size(B, 1);
   m = sqrt(real(n));

   A = reshape(B, [m, m]);

end % function GetDensityMat

% Hilbert space to Liouville space
function SetDensityL(this, densityH)
   this.DensityL = densityH(:); %reshape(densityH, [4*4, 1]);
end % function GetDensityMat

function L0ToNormalBasis(this)
   B = this.DensityL;
   %this.DensityL = this.Hamiltonian.H0ToNormalBasisDensity(B);
   this.DensityL = ctranspose(this.Hamiltonian.RightEigenvectorL)*  B; % round
end

function NormalToL0Basis(this)
   B = this.DensityL;
   %this.DensityL = this.Hamiltonian.NormalToH0BasisDensity(B);
   this.DensityL =  this.Hamiltonian.RightEigenvectorL* B;
end

function HToSzIzBasis(this)
   B = GetDensityMat(this);
   Density = this.Hamiltonian. H0ToSzIzBasisH(B);
   SetDensityL(this, Density);
end
```

```
      function SzIzToHbasis(this)
         B=GetDensityMat(this);
         Density = this.Hamiltonian.SzIzToH0BasisH(B);
         SetDensityL(this, Density);
      end

   end % methods

end %classdef Density_class
```

3. FED_class

```
%Free Evolution Decay (FED)
% Calculate FEDOperator, and Density evolve under a FEDOperator

classdef FED_class < handle

   properties
      Hamiltonian; % class
      Density; % class

   end %properties

   methods

      % constructor
      function this = FED_class(hamiltonian, density)
         this.Density = density;
         this.Hamiltonian = hamiltonian;

      end %  FED_class constructor this

      % Calculate FED superoperator in Liouville space
      function [fEDOperator] = CalFEDOperator(this, t)
         liouville = this.Hamiltonian.LiouvilleDig;
         fEDOperator = expm(t * liouville);
         end % CalFEDOperator


      % Calculate density evolve under a FEDOperator
      function DensityEvolve(this, t)
         % t, time of Free Evolution Decay

         fEDOperator = this.CalFEDOperator(t);
         this.Density.DensityL = fEDOperator * this.Density.DensityL;

      end % DensityEvolve
```

```
    end %method
end %classdef FED_class
```

4. FluctuationRelaxation_class

```
classdef FluctuationRelaxation_class < handle
    properties
        HamiltonianFactor;

        Relaxation = zeros(16, 16); % Spin-lattice and fluctuatin Relaxation operator in Liouville
space and |SzIz> basis
        RelaxationSpin = zeros(16, 16); %  Spin-lattice and spin-spin Relaxation operator in
Liouville space and |SzIz> basis
        RelaxationFluctuation = zeros(16, 16); % Fluctuatin Relaxation operator in Liouville space
and |SzIz> basis

        T2e = 1.0; % Off-diagonal electron spin–spin relaxation, unit: microsecond
        T2n = 1.0; % Off-diagonal nuclear spin–spin relaxation, unit: microsecond

        We = -1.0; % Diagonal electron spin–spin relaxation, unit: microsecond
        Wn = -1.0; % Diagonal nuclear spin–spin relaxation, unit: microsecond

        Wx = -1.0; % Diagonal electron nuclear spin–spin relaxation, unit: microsecond
        Wy = -1.0; % Diagonal electron nuclear spin–spin relaxation, unit: microsecond

        Whe = -1.0; % Heisenberg exchange relaxation

        % For calculation of orientation-dependent spin relaxation matrix [LFP Appendix B]
        % The fluctuating time-dependent perturbation leading
        % to the relaxation can be written in general form as follows:
        % HI = Sum(FiAi)
        F = zeros(10,1); % the time-dependent fluctuating parts of the Hamiltonian,
        % which are defined in the molecular frame and transformed
        % into the lab frame.
        AMatrix = zeros(10, 10); % the spin operators in the lab frame

        Fluctuation = 0.01; % the factor of The fluctuating time-dependent perturbation
        Tc = 1.0; %the correlation time, unit:microsecond, at page 2322 [Freed, 1965]

        Jacac = 0.d0;
        Jbdbd = 0.d0;
        Jadad = 0.d0;
        Jbcbc = 0.d0;
        Jabab = 0.d0;
        Jcdcd = 0.d0;
        Jaabb = 0.d0;
        Jaacc = 0.d0;
        Jaadd = 0.d0;
        Jbbcc = 0.d0;
        Jbbdd = 0.d0;
        Jccdd = 0.d0;

        Jaaaa = 0.d0;
        Jbbbb = 0.d0;
        Jcccc = 0.d0;
        Jdddd = 0.d0;

    end %properties

    methods

        %Suppose, at initial state, the spin is in in z-axis
        function this = FluctuationRelaxation_class(hamiltonianFactor, t2e, t2n, we, wn, wx, wy,
whe, fluctuation, tc)

            this.HamiltonianFactor = hamiltonianFactor;

            %relaxation = zeros(16, 16); % Relaxation operator in Liouville space and |SzIz> basis

            this.T2e = t2e; % T2 electron spin–spin relaxation, unit: microsecond
            this.T2n = t2n; % T2 nuclear spin–spin relaxation, unit: microsecond

            this.We = we; % Diagonal electron spin–spin relaxation, unit: microsecond
            this.Wn = wn;% Diagonal nuclear spin–spin relaxation, unit: microsecond

            this.Wx = wx;% Diagonal electron nuclear spin–spin relaxation, unit: microsecond
            this.Wy = wy; % Diagonal electron nuclear spin–spin relaxation, unit: microsecond

            this.Whe = whe; % Heisenberg exchange relaxation

            this.Fluctuation = fluctuation;
            this.Tc = tc; % the correlation time, unit:microsecond, at page 2322 [Freed, 1965]

            this.InitFluctuating();

            this.Relaxation = this.RelaxationFluctuation;

        end %  Relaxation_class constructor this

        % LPF's table 4
        function InitFluctuating(this)
            % tc the correlation time

            fluctuation = this.Fluctuation;

            dg = fluctuation;
            da = fluctuation;
            df = fluctuation;
            df2 = fluctuation;
            dd = fluctuation;
            dd2 = fluctuation;

            this.FluctuatingFactor(dg, da, df, df2, dd, dd2);
            this.SpinMatrix();
            this.FluctuatingMatrix();
            this.FluctuatingRelaxation();

            this.RelaxationFluctuation = -this.RelaxationFluctuation; % compatible with the LPF
A22 and A24

            %this.FluctuatingTime();
        end

        function FluctuatingRelaxation(this)

            raabb = 2*this.Jabab;
            raacc = 2*this.Jacac;
            raadd = 2*this.Jadad;

            rbbaa = raabb;
            rbbcc = 2*this.Jbcbc;
            rbbdd = 2*this.Jbdbd;

            rccaa = raacc;
            rccbb = rbbcc;
            rccdd = 2*this.Jcdcd;

            rddaa = raadd;
            rddbb = rbbdd;
            rddcc = rccdd;

            raaaa = -(raabb + raacc + raadd);
            rbbbb = -(rbbaa + rbbcc + rbbdd);
            rcccc = -(rccaa + rccbb + rccdd);
            rdddd = -(rddaa + rddbb + rddcc);
```

```
rabab = 2*this.Jaabb - this.Jaaaa - this.Jbbbb + (raaaa + rbbbb)/2;
racac = 2*this.Jaacc - this.Jaaaa - this.Jcccc + (raaaa + rcccc)/2;
radad = 2*this.Jaadd - this.Jaaaa - this.Jdddd + (raaaa + rdddd)/2;

rbaba = rabab;
rbcbc =  2*this.Jbbcc - this.Jbbbb - this.Jcccc + (rbbbb + rcccc)/2;
rbdbd =  2*this.Jbbdd - this.Jbbbb - this.Jdddd + (rbbbb + rdddd)/2;

rcaca = racac;
rcbcb = rbcbc;
rcdcd =  2*this.Jccdd - this.Jcccc - this.Jdddd + (rcccc + rdddd)/2;

rdada = radad;
rdbdb = rbdbd;
rdcdc = rcdcd;

relaxation = zeros(16, 16); % Relaxation operator in Liouville space and |SzIz> basis

relaxation(1,1) = raaaa;
relaxation(2,2) = rabab;
relaxation(3,3) = racac;
relaxation(4,4) = radad;
relaxation(5,5) = rbaba;
relaxation(6,6) = rbbbb;
relaxation(7,7) = rbcbc;
relaxation(8,8) = rbdbd;
relaxation(9,9) = rcaca;
relaxation(10,10) = rcbcb;
relaxation(11,11) = rcccc;
relaxation(12,12) = rcdcd;
relaxation(13,13) = rdada;
relaxation(14,14) = rdbdb;
relaxation(15,15) = rdcdc;
relaxation(16,16) = rdddd;

relaxation(6,1) = rbbaa;
relaxation(11,1) = rccaa;
relaxation(16,1) = rddaa;

relaxation(1,6) = raabb;
relaxation(11,6) = rccbb;
relaxation(16,6) = rddbb;

relaxation(1,11) = raacc;
relaxation(6,11) = rbbcc;
relaxation(16,11) = rddcc;
```

```
    relaxation(1,16) = raadd;
    relaxation(6,16) = rbbdd;
    relaxation(11,16) = rccdd;

    this.RelaxationFluctuation = relaxation;
end % FluctuatingRelaxation

% Orientation-dependent spin relaxation matrix
% From Slitcher Eq. (5.294)
function FluctuatingMatrix(this)
    tc = this.Tc; % tc correlation time, unit: microsecond.

    ea = this.HamiltonianFactor.Ea;
    eb = this.HamiltonianFactor.Eb;
    ec = this.HamiltonianFactor.Ec;
    ed = this.HamiltonianFactor.Ed;

    wac = ea - ec;
    wbd = eb - ed;
    wad = ea - ed;
    wbc = eb - ec;
    wab = ea - eb;
    wcd = ec - ed;

    jacac = 0.d0;
    jbdbd = 0.d0;
    jadad = 0.d0;
    jbcbc = 0.d0;
    jabab = 0.d0;
    jcdcd = 0.d0;

    jaabb = 0.d0;
    jaacc = 0.d0;
    jaadd = 0.d0;
    jbbcc = 0.d0;
    jbbdd = 0.d0;
    jccdd = 0.d0;

    jaaaa = 0.d0;
    jbbbb = 0.d0;
    jcccc = 0.d0;
    jdddd = 0.d0;

    for i = 1:10
        jacac = jacac + this.F(i)* this.F(i)* this.AMatrix(i,1);
        jbdbd = jbdbd + this.F(i)* this.F(i)* this.AMatrix(i,2);
        jadad = jadad + this.F(i)* this.F(i)* this.AMatrix(i,3);
        jbcbc = jbcbc + this.F(i)* this.F(i)* this.AMatrix(i,4);
        jabab = jabab + this.F(i)* this.F(i)* this.AMatrix(i,5);
        jcdcd = jcdcd + this.F(i)* this.F(i)* this.AMatrix(i,6);

        jaabb = jaabb + this.F(i)* this.F(i)* sqrt(this.AMatrix(i,7) * this.AMatrix(i,8));
        jaacc = jaacc + this.F(i)* this.F(i)* sqrt(this.AMatrix(i,7) * this.AMatrix(i,9));
        jaadd = jaadd + this.F(i)* this.F(i)* sqrt(this.AMatrix(i,7) * this.AMatrix(i,10));
        jbbcc = jbbcc + this.F(i)* this.F(i)* sqrt(this.AMatrix(i,8) * this.AMatrix(i,9));
        jbbdd = jbbdd + this.F(i)* this.F(i)* sqrt(this.AMatrix(i,8) * this.AMatrix(i,10));
        jccdd = jccdd + this.F(i)* this.F(i)* sqrt(this.AMatrix(i,9) * this.AMatrix(i,10));

        jaaaa = jaaaa + this.F(i)* this.F(i)* this.AMatrix(i,7);
        jbbbb = jbbbb + this.F(i)* this.F(i)* this.AMatrix(i,8);
        jcccc = jcccc + this.F(i)* this.F(i)* this.AMatrix(i,9);
        jdddd = jdddd + this.F(i)* this.F(i)* this.AMatrix(i,10);
    end

    this.Jacac = tc/(1+(wac*tc)^2)*jacac;
    this.Jbdbd = tc/(1+(wbd*tc)^2)*jbdbd;
    this.Jadad = tc/(1+(wad*tc)^2)*jadad;
    this.Jbcbc = tc/(1+(wbc*tc)^2)*jbcbc;
    this.Jabab = tc/(1+(wab*tc)^2)*jabab;
    this.Jcdcd = tc/(1+(wcd*tc)^2)*jcdcd;

    this.Jaabb = tc*jaabb;
    this.Jaacc = tc*jaacc;
    this.Jaadd = tc*jaadd;
    this.Jbbcc = tc*jbbcc;
    this.Jbbdd = tc*jbbdd;
    this.Jccdd = tc*jccdd;

    this.Jaaaa = tc*jaaaa;
    this.Jbbbb = tc*jbbbb;
    this.Jcccc = tc*jcccc;
    this.Jdddd = tc*jdddd;

end% FluctuatingMatrix

% Table 4 of LPF
function FluctuatingFactor(this, dg, da, df, df2, dd, dd2)
    % dg, da, df, df2, dd, dd2, the interaction constants
    % are the time-dependent deviations from the average values
    % [e.g., dg = g(t)-gAverage.
```

```
        %bohr = 9.27400968D-28;  %Bohr magneton: ß_e,µ_B=9.27400968(20)×10^(-28)
J*Gauss^(-1)
        %planck = 6.62606957D-34; %Planck constant: h=6.62606957(29)×10^(-34) J*S

        gyromagneticRatio = 2.802495266D0; %Electron gyromagnetic ratio: r_e/2pi=2.8024952
66(62)MHz*Gauss^(-1)
        %gyromagneticRatio_h = 42.576D0; %Nuclear gyromagnetic ratio:
r_h/2pi=42.576MHz*Gauss^(-1)

        euler_angles = this.HamiltonianFactor.Euler_angles; % !euler angles of the rotating
frame, unit: rad

        f0 = this.HamiltonianFactor.F0; % uint: MHz

        gAverage = this.HamiltonianFactor.GAverage; % unit: none
        aAverage = this.HamiltonianFactor.AAverage; % unit: Gauss

        f = this.HamiltonianFactor.F; % uint: MHz
        d = this.HamiltonianFactor.D; % unit: Gauss

        f2 = this.HamiltonianFactor.F2; % unit: Gauss
        d2 = this.HamiltonianFactor.D2; % unit: Gauss

        % alpha = euler_angles(1);
        beta = euler_angles(2);
        gamma = euler_angles(3);

        %tcf = 20/(f0*sqrt(3)); % At page 2322 [Freed, 1965]
        %disp(tcf)

        gAverage = dg*gAverage;
        aAverage = da*aAverage;
        f  = df*f;
        f2 = df2*f2;
        d  = dd*d;
        d2 = dd2*d2;

        iso_g = gAverage*f0; % isotropic g
        this.F(1) = iso_g;

        ani_g1 = (1/2)*f * (3*cos(beta)*cos(beta) - 1) + f2 * sin(beta)*sin(beta) * cos(2*
gamma); %Anisotropic g
        this.F(2) = ani_g1;

        ani_g2 = (3/8)*f*sin(2*beta) - (1/4)*f2*sin(2*beta) * cos(2* gamma);
        this.F(3) = ani_g2;
```

```
         iso_hf1 = -gyromagneticRatio * aAverage; % Isotropic hf
         this.F(4) = iso_hf1;

         iso_hf2 = iso_hf1;
         this.F(5) = iso_hf2;

         ani_hf1 = (1/2)*d *(3.0*cos(beta)*cos(beta) - 1) - d2 * sin(beta)*sin(beta) * cos(2 *
gamma); % Anisotropic hf
         ani_hf1 = -gyromagneticRatio * ani_hf1;
         this.F(6) = ani_hf1;

         ani_hf2 = ani_hf1;
         this.F(7) = ani_hf2;

         ani_hf3 = (3/8)*d*sin(2*beta) - (1/4)*d2 * sin(2*beta) * cos(2* gamma); % Anisotropic
hf
         ani_hf3 = -gyromagneticRatio * ani_hf3;
         this.F(8) = ani_hf3;

         ani_hf4 = ani_hf3;
         this.F(9) = ani_hf4;

         ani_hf5 = (3/8)*d*sin(beta)*sin(beta) - (1/4)*d2 * (1 + cos(beta)*cos(beta)) * cos(2*
gamma); % Anisotropic hf
         ani_hf5 = -gyromagneticRatio * ani_hf5;
         this.F(10) = ani_hf5;

      end % function FluctuatingFactor

      % the spin operators in the lab frame
      function SpinMatrix(this)

         m1 = this.HamiltonianFactor.M1;
         m2 = this.HamiltonianFactor.M2;

         c1 = this.HamiltonianFactor.C1;
         c2 = this.HamiltonianFactor.C2;
         c3 = this.HamiltonianFactor.C3;
         c4 = this.HamiltonianFactor.C4;

         this.AMatrix(1,7) = 1/4;
         this.AMatrix(1,8) = this.AMatrix(1,7);
         this.AMatrix(1,9) = 1/4;
         this.AMatrix(1,10) = this.AMatrix(1,9);
         this.AMatrix(2,7) = 1/4;
         this.AMatrix(2,8) = this.AMatrix(2,7);
         this.AMatrix(2,9) = 1/4;
         this.AMatrix(2,10) = this.AMatrix(2,9);

         this.AMatrix(3,1) = m1*m1;
         this.AMatrix(3,2) = m1*m1;
         this.AMatrix(3,3) = m2*m2;
         this.AMatrix(3,4) = m2*m2;

         this.AMatrix(4,5) = (1/4)*real(conj(c1)*c2)*real(conj(c1)*c2);
         this.AMatrix(4,6) = (1/4)*real(conj(c3)*c4)*real(conj(c3)*c4);
         this.AMatrix(4,7) = (1/16)*(abs(c1)*abs(c1) - abs(c2)*abs(c2));
         this.AMatrix(4,8) = this.AMatrix(4,7);
         this.AMatrix(4,9) = (1/16)*(abs(c3)*abs(c3) - abs(c4)*abs(c4));
         this.AMatrix(4,10) = this.AMatrix(4,9);


         this.AMatrix(5,1) = (1/4)*abs(conj(c2)*c3)*abs(conj(c2)*c3);
         this.AMatrix(5,2) = (1/4)*abs(conj(c1)*c4)*abs(conj(c1)*c4);
         this.AMatrix(5,3) = (1/4)*abs(conj(c2)*c4)*abs(conj(c2)*c4);
         this.AMatrix(5,4) = (1/4)*abs(conj(c1)*c3)*abs(conj(c1)*c3);

         this.AMatrix(6,5) = (1/4)*real(conj(c1)*c2)*real(conj(c1)*c2);
         this.AMatrix(6,6) = (1/4)*real(conj(c3)*c4)*real(conj(c3)*c4);
         this.AMatrix(6,7) = (1/16)*(abs(c1)*abs(c1) - abs(c2)*abs(c2));
         this.AMatrix(6,8) = this.AMatrix(6,7);
         this.AMatrix(6,9) = (1/16)*(abs(c3)*abs(c3) - abs(c4)*abs(c4));
         this.AMatrix(6,10) = this.AMatrix(6,9);


         this.AMatrix(7,1) = (1/16)*abs(conj(c2)*c3)*abs(conj(c2)*c3);
         this.AMatrix(7,2) = (1/16)*abs(conj(c1)*c4)*abs(conj(c1)*c4);
         this.AMatrix(7,3) = (1/16)*abs(conj(c2)*c4)*abs(conj(c2)*c4);
         this.AMatrix(7,4) = (1/16)*abs(conj(c1)*c3)*abs(conj(c1)*c3);

         this.AMatrix(8,5) = (1/4)*(abs(c1)*abs(c1) - abs(c2)*abs(c2))^2;
         this.AMatrix(8,6) = (1/4)*(abs(c3)*abs(c3) - abs(c4)*abs(c4))^2;
         this.AMatrix(8,7) = real(conj(c1)*c2)^2;
         this.AMatrix(8,8) = this.AMatrix(8,7);
         this.AMatrix(8,9) = real(conj(c3)*c4)^2;
         this.AMatrix(8,10) = this.AMatrix(8,9);

         this.AMatrix(9,1) = (1/4)*(abs(conj(c1)*c3-conj(c2)*c4))^2;
         this.AMatrix(9,2) = (1/4)*(abs(conj(c1)*c3-conj(c2)*c4))^2;
         this.AMatrix(9,3) = (1/4)*(abs(conj(c1)*c4+conj(c2)*c3))^2;
```

```
            this.AMatrix(9,4) = (1/4)*(abs(conj(c1)*c4+conj(c2)*c3))^2;

            this.AMatrix(10,1) = abs(conj(c1)*c4)^2;
            this.AMatrix(10,2) = abs(conj(c2)*c3)^2;
            this.AMatrix(10,3) = abs(conj(c1)*c3)^2;
            this.AMatrix(10,4) = abs(conj(c2)*c4)^2;
        end % function

        function FluctuatingTime(this)
            gyromagneticRatio_h = 42.576D0; %Nuclear gyromagnetic ratio:
r_h/2pi=42.576MHz*Gauss^(-1)

            w0 = this.HamiltonianFactor.W0; % uint: MHz

            tcs = 20/(sqrt(3)*w0);
            fprintf('[Freed 1965] tc = %d microseconds\n',tcs);

            for i = 1:10 % Slitcher  Eq. (5.301)
                tcs = (gyromagneticRatio_h*this.F(i)/w0)^2;
                tcs = tcs/(3*this.We);
                fprintf('[Slichter 2013] %uth tc = %d microseconds\n',i, tcs);
            end

            %               for i = 1:10 % Slitcher  Eq. (5.301)
            %                   b = (gyromagneticRatio_h*this.F(i))^2;
            %                   b = b/(3*this.We);
            %                   tcs = b + sqrt(b*b - 4*w0*w0);
            %                   tcs = tcs/2;
            %                   disp(tcs)
            %               end

            fprintf('w0 = %d \n', w0);
            fprintf('gramma_n = %d \n', gyromagneticRatio_h);
                        for i = 1:10 % Slitcher  Eq. (5.301)
                            t1 = (gyromagneticRatio_h*this.F(i))^2;
                            t1 = 3*w0/t1;
                            fprintf('[Slichter 2013] fp^2 = %d, t1 = %d \n', this.F(i)^2, t1);
                        end

        end %function FluctuatingTime

    end % methods

end %classdef FluctuationRelaxation_class
```

5. Hamiltonian_class

```
% Calculate Hamiltonian matrix, LPF Eq. 3
classdef Hamiltonian_class < handle
    properties
        HamiltonianFactor;
        Relaxation; % LPF A22
        FluctuationRelaxation; % fluctuation relaxation
        RelaxationType = 1;

        HamiltonianH = zeros(4,4); % Hamiltonian operator in Hilbert space and |SzIz> basis
        EigenvalueH = zeros(4,1); % Of Hamiltonian operator.
        RightEigenvectorH; % Of Hamiltonian operator. Au = tu, any such column vector u is
called a right eigenvector of A

        RightEigenvectorL;

        LiouvilleDig = zeros(16,16); % diagonal matrix of generalized eigenvalues

    end %properties

    methods
        % constructor
        function this = Hamiltonian_class(hamiltonianFactor, relaxation, fluctuationRelaxation,
relaxationType)
            this.HamiltonianFactor = hamiltonianFactor;
            this.Relaxation = relaxation;
            this.FluctuationRelaxation = fluctuationRelaxation;
            this.RelaxationType = relaxationType;
        end % end constructor

        function Init(this, euler_angles)
            hamiltonianFactor = this.HamiltonianFactor;
            hamiltonianFactor.Init(euler_angles);

            wn = hamiltonianFactor.Wn; % the nuclear Larmor frequency for the nucleus, uint: MHz
            C = hamiltonianFactor.C; %uint: MHz
            A = hamiltonianFactor.A; %uint: MHz
            B1 = hamiltonianFactor.B1; %uint: MHz
            B2 = hamiltonianFactor.B2; %uint: MHz

            % Initialize Hamiltonian operator matrix in Hilbert space
            hamiltonianH = zeros(4,4);
            hamiltonianH(1,1) = 0.5*(C - wn + 0.5*A);
            hamiltonianH(2,2) = 0.5*(C + wn - 0.5*A);
            hamiltonianH(3,3) = 0.5*(-C - wn - 0.5*A);
            hamiltonianH(4,4) = 0.5*(-C + wn + 0.5*A);

            hamiltonianH(1,2) = 0.25*B1;
            hamiltonianH(2,1) = 0.25*B2;

            hamiltonianH(3,4) = -0.25*B1;
            hamiltonianH(4,3) = -0.25*B2;

            this.HamiltonianH = hamiltonianH;

            % Diagonalize the Hamiltonian operator matrix in Hilbert space
            %[V,D,W] = eig(A)
            % D: diagonal matrix of generalized eigenvalues
            % V: Right eigenvectors: A*V = V*D, % Au = tu, Any such column vector u is called a
right eigenvector of A
            % W: Left eigenvectors: W'*A = D*W
            [V, D] = eig(hamiltonianH, 'vector');

            % Swap ascending order to descending order
            eigenvalueH = D;
            eigenvalueH([1,3],:) = eigenvalueH([3,1],:);
            %eigenvalueH([1,4],:) = eigenvalueH([4,1],:);
            %eigenvalueH([2,3],:) = eigenvalueH([3,2],:);

            this.EigenvalueH = eigenvalueH;

            rightEigenvectorH = V;
            rightEigenvectorH(:,[1,3]) = rightEigenvectorH(:,[3,1]);
            %rightEigenvectorH(:,[1,4]) = rightEigenvectorH(:,[4,1]);
            %rightEigenvectorH(:,[2,3]) = rightEigenvectorH(:,[3,2]);

            this.RightEigenvectorH = rightEigenvectorH; % It is the change of basis matrix from
hamiltonianH (original) to D (new).

            rndoff = 10e12;
            h0 = diag(this.EigenvalueH);

            relaxationType = this.RelaxationType;
            switch relaxationType
              case 1 % LPF A22
                relaxation0 = this.Relaxation.Relaxation;
              case 2 % fluctuation relaxation
                relaxation0 = this.FluctuationRelaxation.RelaxationFluctuation;
              case 3 % LPF A22 and fluctuation relaxation
                relaxation0 = this.Relaxation.Relaxation + this.FluctuationRelaxation.Relaxation;
              otherwise
```

```
                    error('1, 2 or 3 relaxation type only!')
            end %switch relaxationType

            liouville0 = kron(eye(4),h0) -  kron(transpose(h0), eye(4));
            liouville = round(rndoff*(-1i*liouville0 + relaxation0))/rndoff; % compatible with the
LPF A22 and A24

            [V1, D1] = eig(liouville);
            rightEigenvectorL = V1;
            this.RightEigenvectorL = rightEigenvectorL;
            this.LiouvilleDig  = round(D1*rndoff)/rndoff;

        end % Hamiltonian_class constructor

        % Wihtout the relaxation part H = H0
        function [A] = SzIzToH0BasisH (this,operatorH)
            A = ctranspose(this.RightEigenvectorH)* operatorH * this.RightEigenvectorH;
        end

        function [A] = H0ToSzIzBasisH (this, operatorH)
            A = this.RightEigenvectorH * operatorH * ctranspose(this.RightEigenvectorH);
        end

        % Wiht the relaxation part H = H0 + R
        function [A] = L0ToNormalBasis (this, operatorL)
            A = ctranspose(this.RightEigenvectorL)* operatorL * this.RightEigenvectorL;
        end

    end % methods

end %classdef Hamiltonian_class
```

6. HamiltonianFactor_class

```
% Calculate Hamiltonian parameters, LPF Eqs.2 and 4

classdef HamiltonianFactor_class < handle
    properties
        B0; % static magnetic field, unit: Gauss
        Wn; % the nuclear Larmor frequency for the nucleus, uint: MHz
        G_tensor; % g_tensor(3)
        Hyperfine_tensor; % hyperfine_tensor(3) !hyperfine tensor, unit: Gauss

        Euler_angles = zeros(3);

        C = 0.0d0;     %uint: MHz
        A = 0.0d0;     %uint: MHz
        B1 = 0.0d0;    %uint: MHz
        B2 = 0.0d0;    %uint: MHz

        W0 = 0.0d0;    % the electron Larmor frequency for the nucleus, uint: MHz
        % For calculation of orientation-dependent spin relaxation matrix

        GAverage  = 0.0d0; % unit: none
        AAverage  = 0.0d0; % unit: Gauss

        F0 = 0.0d0; % uint: MHz

        F = 0.0d0;  % uint: MHz
        F2 = 0.0d0; %  uint: MHz

        D = 0.0d0;  % unit: Gauss
        D2 = 0.0d0; % unit: Gauss

        Wa = 0.0d0;    %uint: MHz
        Wb = 0.0d0;    %uint: MHz

        C1 = 0.0d0;
        C2 = 0.0d0;
        C3 = 0.0d0;
        C4 = 0.0d0;

        M1 = 0.0d0;
        M2 = 0.0d0;

        Ea = 0.0d0; %uint: MHz
        Eb = 0.0d0; %uint: MHz
        Ec = 0.0d0; %uint: MHz
        Ed = 0.0d0; %uint: MHz
    end %properties

    methods
        function this = HamiltonianFactor_class(B0, wn, g_tensor, hyperfine_tensor, euler_angles)
            % B0 !static magnetic field, unit: Gauss
            % wn !the nuclear Larmor frequency for the nucleus, uint: MHz
            % g_tensor(3)
            % hyperfine_tensor(3) !hyperfine tensor, unit: Gauss
            this.B0 = B0;
            this.Wn = wn;
            this.G_tensor = g_tensor;
            this.Hyperfine_tensor = hyperfine_tensor;
            this.Euler_angles = euler_angles;
```

```
            this.Init(euler_angles)
        end % HamiltonianFactor_class constructor

        function Init(this, euler_angles)
            % euler_angles(3) !euler angles of the rotating frame, unit: rad
            this.Euler_angles = euler_angles;

            bohr = 9.27400968D-28;  %Bohr magneton: ß_e,µ_B=9.27400968(20)×10^(-28)
J*Gauss^(-1)
            planck = 6.62606957D-34; %Planck constant: h=6.62606957(29)×10^(-34) J*S
            %gyromagneticRatio = 2.802495266D0; %Electron gyromagnetic ratio: r_e=2.8024952
66(62)MHz*Gauss^(-1)
            wn = this.Wn;
            g_tensor = this.G_tensor;
            hyperfine_tensor = this.Hyperfine_tensor;

            f0 = (bohr * this.B0 / planck) * 1.0D-6;  % uint: MHz*rad
            this.F0 = f0;

            gAverage =(1.0/3.0)*(g_tensor(1) + g_tensor(2) + g_tensor(3)) * f0;
            this.GAverage = gAverage;
            this.W0 = gAverage; % the electron Larmor frequency for the nucleus, uint: MHz

            aAverage = (1.0/3.0)*(hyperfine_tensor(1) + hyperfine_tensor(2) +
hyperfine_tensor(3)); %unit: Gauss
            this.AAverage = aAverage;

            f = sqrt(2.0/3.0)*(g_tensor(3) - 0.5*(g_tensor(1) + g_tensor(2))) * f0;
            this.F = f; % uint: MHz

            d = (2.0/3.0)*(hyperfine_tensor(3) - 0.5*(hyperfine_tensor(1) + hyperfine_tensor(2)));
            this.D = d;

            f2 = 0.5*(g_tensor(1) - g_tensor(2)) * f0;
            this.F2 = f2;

            d2 = 0.5*(hyperfine_tensor(1) - hyperfine_tensor(2));
            this.D2 = d2;

            % alpha = euler_angles(1);
            beta = euler_angles(2);
            gamma = euler_angles(3);

            this.C = gAverage  + 0.5*f * (3*cos(beta)*cos(beta) - 1) + f2 * sin(beta)*sin(beta) *
cos(2* gamma);
```

```
            this.A = -2*pi*( aAverage + (0.5*d *(3.0*cos(beta)*cos(beta) - 1) + d2 *
sin(beta)*sin(beta) * cos(2 * gamma)));

            this.B1 = -4*pi*(+(3.0/4.0)*d * sin(beta) * cos(beta) - 0.5*d2 * sin(beta)...
            * (cos(beta) * cos(2 * gamma) -1i*sin(2*gamma)));

            this.B2 =  -4*pi*(+(3.0/4.0)*d * sin(beta) * cos(beta) - 0.5*d2 * sin(beta)...
            * (cos(beta) * cos(2 * gamma) +1i*sin(2*gamma)));

            % For calculation of orientation-dependent spin relaxation matrix
            wa = (this.A/2- wn)^2 + (abs(this.B1/2))^2;
            wa = sqrt(wa);
            this.Wa = wa;

            wb = (this.A/2 + wn)^2 + (abs(this.B1/2))^2;
            wb = sqrt(wb);
            this.Wb = wb;

            this.C1 = 1 + (this.A/2 - wn)/wa;
            this.C1 = sqrt(this.C1/2);

            this.C2 = 1 - (this.A/2 - wn)/wa;
            this.C2 = -sqrt(this.C2/2);

            this.C3 = 1 + (this.A/2 + wn)/wb;
            this.C3 = sqrt(this.C3/2);

            this.C4 = 1 - (this.A/2 + wn)/wb;
            this.C4 = -sqrt(this.C4/2);

            this.M1 = 1 - (wn^2 - (this.A/2)^2 - (abs(this.B1/2)^2))/(wa*wb);
            this.M1 = sqrt(this.M1/2);

            this.M2 = 1 + (wn^2 - (this.A/2)^2 - (abs(this.B1/2)^2))/(wa*wb);
            this.M2 = sqrt(this.M2/2);

            %LPF' Eq.(A3)
            this.Ea = this.C/2 + wa/2;
            this.Eb = this.C/2 - wa/2;
            this.Ec = -this.C/2 - wb/2;
            this.Ed = -this.C/2 + wb/2;
        end % function Init

    end % methods

end %classdef HamiltonianFactor_class
```

7. Pathway_class

```
% Coherence Pathway of Dr. Misra's DQC

classdef Pathway_class < handle

    properties
        Density; % class
        Hamiltonian;
        P1;
        P0;
        P_1;
    end %properties

    methods
        % constructor
        function this = Pathway_class(density,hamiltonian)
            this.Density = density;
            this.Hamiltonian=hamiltonian;
            splus = zeros(2,2); % s+ of 1/2 spin
            splus(1,2) = 1;

            sminus = zeros(2,2);% s- of 1/2 spin
            sminus(2,1) = 1;

            this.P1 = kron(splus, ones(2));
            this.P0 = kron(eye(2), ones(2));
            this.P_1 = kron(sminus, ones(2));
        end % Pathway_class constructor

        function SelectPath(this, spinPathway)

            % In SzIz basis
            densityH = this.Density.GetDensityMat();

           switch spinPathway
                case 1 % p=1
                    densityH = (this.P1).* densityH;
                case 0 % p=0
                    densityH = (this.P0).* densityH;
                case -1 % p=-1
                    densityH = (this.P_1).* densityH;
                case -11 % p=-1 or 1
                    densityH = (this.P1 + this.P_1).* densityH;
                otherwise
                    error('Error. \nNo such pathway %d.', spinPathway)
            end %switch path

            this.Density.SetDensityL(densityH);

        end % SelectPath

    end % methods

end % Pathway_class < handle
```

8. Plot_class

```
% FFT and plot

classdef Plot_class < handle

    properties
        Fs1 = 64;           % Sampling frequency
        Dt1;          % Sample time, unit: Microsecond
        N1;           % Length of signal
        Tv1;   % Time vector, unit: Microsecond

        Df1;          % Frequency increment, unit: MHz
        Fv1; % Frequency vector, unit: MHz

        Scaling_zfft;      % Scaling ratio of intensity after fft

        Fs2 = 64;           % Sampling frequency
        Dt2;          % Sample time, unit: Microsecond
        N2;           % Length of signal
        Tv2;   % Time vector, unit: Microsecond

        Df2;          % Frequency increment, unit: MHz
        Fv2; % Frequency vector, unit: MHz

        Signal2d;    % Sampled data
        Signal2d_fft;    % Discrete Fourier transform (DFT)
    end %properties

    methods

        % constructor
        function this = Plot_class(fs1, fs2)
            this.Init(fs1, fs2);
        end %  ESEEM_LS_FFT_class constructor this

        function Init(this, fs1, fs2)
            %fs1 = 64;                  % Sampling frequency
            dt1 = 1/fs1;               % Sample time, unit: Microsecond
            n1 = 1*fs1;                 % Length of signal
            tv1 = (0:n1-1)*dt1; % (-n1/2:n1/2-1)*dt1;   % Time vector, unit: Microsecond

            df1 = fs1/n1;               % Frequency increment, unit: MHz
            fv1 = (-n1/2:n1/2-1)*(df1); % Frequency vector, unit: MHz

            scaling_zfft = dt1;          % Scaling ratio of intensity after fft

            %fs2 = 64;                  % Sampling frequency
            dt2 = 1/fs2;               % Sample time, unit: Microsecond
            n2 = 1*fs2;                 % Length of signal
            tv2 = (0:n2-1)*dt2;%(-n2/2:n2/2-1)*dt2;   % Time vector, unit: Microsecond

            df2 = fs2/n2;               % Frequency increment, unit: MHz
            fv2 = (-n2/2:n2/2-1)*(df2); % Frequency vector, unit: MHz

            signal2d = zeros(n1,n2);    % Sampled data

            this.Fs1 = fs1;                  % Sampling frequency
            this.Dt1 =  dt1;                % Sample time, unit: Microsecond
            this.N1 =  n1;                 % Length of signal
            this.Tv1 = tv1;   % Time vector, unit: Microsecond

            this.Df1 = df1;               % Frequency increment, unit: MHz
            this.Fv1 = fv1; % Frequency vector, unit: MHz

            this.Scaling_zfft = scaling_zfft;          % Scaling ratio of intensity after fft

            this.Fs2 = fs2;                  % Sampling frequency
            this.Dt2 = dt2;                % Sample time, unit: Microsecond
            this.N2 = n2;                 % Length of signal
            this.Tv2 = tv2;   % Time vector, unit: Microsecond

            this.Df2 = df2;               % Frequency increment, unit: MHz
            this.Fv2 = fv2; % Frequency vector, unit: MHz

            this.Signal2d = signal2d;    % Sampled data

        end %  Init this

        function SignalToFFT(this, signalFile)

            signal2d = dlmread(signalFile); % Read in data
```

```
            withFilter = false; %false;
            if withFilter == true
                fst = 0.15; %Fst — frequency at the end of the stop band. Specified in normalized
frequency units. Also called Fstop.
                fp = 0.25; %Fp — frequency at the start of the pass band. Specified in normalized
frequency units. Also called Fpass.
                ast = 60; %Ast — attenuation in the stop band in decibels (the default units). Also
called Astop.
                ap = 1; %Ap — amount of ripple allowed in the pass band in decibels (the default
units). Also called Apass.

                d = fdesign.highpass('Fst,Fp,Ast,Ap',fst,fp,ast,ap);
                Hd = design(d,'equiripple');
                signal2d = filter(Hd,signal2d);
            end %if withFilter == true

            this.Signal2d = signal2d;

            signal2d_fft = fft2(signal2d);              % Discrete Fourier transform (DFT)
            signal2d_fft = fftshift(signal2d_fft);      % Shift zero-frequency component to center of
spectrum
            %signal2d_fft = this.Scaling_zfft * signal2d_fft; % Scale output intensity after fft
            this.Signal2d_fft = signal2d_fft;

        end% function SignalToFFT

        function Plot(this, titleName, rotateDegree, plotAll)
            % rotateDegree: rotate the plot around z-axis, unit:degree, not radian

            %fs1 = this.Fs1;    % Sampling frequency
            %dt1 = this.Dt1;    % Sample time, unit: Microsecond
            n1 = this.N1;   % Length of signal
            tv1 = this.Tv1; % Time vector, unit: Microsecond

            %df1 = this.Df1;    % Frequency increment, unit: MHz
            fv1 = this.Fv1; % Frequency vector, unit: MHz

            %scaling_zfft = this.Scaling_zfft;  % Scaling ratio of intensity after fft

            %fs2 = this.Fs2;    % Sampling frequency
            %dt2 = this.Dt2;    % Sample time, unit: Microsecond
            n2 = this.N2;   % Length of signal
            tv2 = this.Tv2; % Time vector, unit: Microsecond

            %df2 = this.Df2;    % Frequency increment, unit: MHz
            fv2 = this.Fv2; % Frequency vector, unit: MHz
```

```
signal2d = this.Signal2d;
signal2d = signal2d /(max(max(abs(signal2d)))); % scale to 1
signal2d = abs(signal2d);

if (~exist('plotAll', 'var'))
    plotAll = false;
end

if plotAll == true
    Figure('Name', strcat('2D-', titleName, '(Time-domain)'), 'Visible', 'On');
    mesh(tv1,tv2, signal2d)
    axis([0 1 0 1 0 1.1])
    axis tight
    xlabel('T1 (\mus)') %Microsecond
    ylabel('T2 (\mus)')
    zlabel('Intensity')
    %title(strcat('{\bf ', '2D-', titleName, ' (Time-domain)', '}'));
    %colormap gray

    row = n1/2;
    col = n2/2;
    y1 = signal2d(1:n1, col);
    y2 = signal2d(row, 1:n2);

    Figure('Name',strcat('1D-', titleName, '(Time-domain)'),'Visible','On');
    plot(tv1,y1)
    title(strcat('{\bf ', '1D-', titleName, ' (Time-domain)', '}'));
    xlabel('T1 (Microsecond)')
    ylabel('Intensity')

    Figure('Name',strcat('1D-', titleName, '(Time-domain)'),'Visible','On');
    plot(tv2,y2)
    title(strcat('{\bf ', '1D-', titleName, ' (Time-domain)', '}'));
    xlabel('T2 (Microsecond)')
    ylabel('Intensity')

    y1_fft = fft(y1,n1);
    y1_fft = fftshift(y1_fft);
    % Plot single-sided amplitude spectrum.
    Figure('Name',strcat('1D-', titleName, '(Frequency-domain)'),'Visible','On');
    plot(fv1,abs(y1_fft))
    title(strcat('{\bf ', '1D-', titleName, ' (Frequency-domain)', '}'));
    xlabel('F1 (MHz)')
    ylabel('Intensity')

    y2_fft = fft(y2,n2);
```

```
        y2_fft = fftshift(y2_fft);
        % Plot single-sided amplitude spectrum.
        Figure('Name',strcat('1D-', titleName, '(Frequency-domain)'),'Visible','On');
        plot(fv2,abs(y2_fft))
        title(strcat('{\bf ', '1D-', titleName, ' (Frequency-domain)', '}'));
        xlabel('F2 (MHz)')
        ylabel('Intensity')

      end

      signal2d_fft = this.Signal2d_fft;
      signal2d_fft = signal2d_fft /(max(max(signal2d_fft))); %Scale to 1
      signal2d_fft = abs(signal2d_fft); % Amplitude of the DFT. real-dispersion, imag-absorption, abs-Complex Magnitude

%       maximum = max(max(signal2d_fft));
%       [row,col]=find(signal2d_fft==maximum)

      signal2d_fft = transpose(signal2d_fft);
      signal2d_fft = this.RotationZ(signal2d_fft, rotateDegree);

      Figure('Name', strcat('2D-', titleName, '(Frequency-domain)'), 'Visible', 'On');
      mesh(fv1, fv2, signal2d_fft)
      xlabel('F1 (MHz)')
      ylabel('F2 (MHz)')
      zlabel('Intensity')
      %title(strcat('{\bf ', '2D-', titleName, ' (Frequency-domain)', '}'));
      set(gca,'XTick',-this.N1/2:50:this.N1/2)
      set(gca,'YTick',-this.N2/2:50:this.N2/2)
      view(45, 22.5);% view(3) sets the default three-dimensional view, az = –37.5, el = 30.
      %signal2d_fft(:,this.N2/2:this.N2/2+2) = 0.0; % remove autopeaks along f1
      %signal2d_fft(this.N1/2:this.N1/2+2,:) = 0.0; % remove autopeaks along f2

     end % function Plot

    function [outmatrix] = RotationZ (~, inmatrix, degree)

      s = sign(degree);
      degree = mod(abs(degree), 360);
      switch degree
        % Special cases
        case 0
          outmatrix = inmatrix;
        case 90
          outmatrix = rot90(inmatrix, s);
        case 180
```

```
                    outmatrix = rot90(inmatrix, s*2);
                case 270
                    outmatrix = rot90(inmatrix, s*3);

                    % General rotations
                otherwise
                    degree = s*degree;

                    [nrows, ncols] = size(inmatrix);
                    midx=ceil((ncols+1)/2);
                    midy=ceil((nrows+1)/2);

                    a = degree*pi/180;
                    Mr = [cos(a) sin(a); -sin(a) cos(a)]; % degree rotation

                    % rotate about center
                    [X, Y] = meshgrid(1:ncols,1:nrows);
                    XYt = [X(:)-midx Y(:)-midy]*Mr;
                    XYt = bsxfun(@plus,XYt,[midx midy]);

                    xout = round(XYt(:,1)); yout = round(XYt(:,2)); % nearest neighbor!
                    %outbound = yout<1 | yout>nrows | xout<1 | xout>ncols;
                    xout(xout<1) = 1; xout(xout>ncols) = ncols;
                    yout(yout<1) = 1; yout(yout>nrows) = nrows;
                    outmatrix = inmatrix(sub2ind(size(inmatrix),yout,xout)); % lookup
                    outmatrix = reshape(outmatrix,size(inmatrix));

            end %switch
        end % function RotationZ

    end %method
end %classdef Plot_class
```

9. Pulse_class

```
% Pulse evolve
% Calculate pulseOperator, and Density evolve under a pulseOperator
classdef Pulse_class  < handle
    properties
        Hamiltonian; % class
        Density; % class

        Pulsetime0 = 0.0; % (~5 ns for pi/2 pulse, LPF p3669) unit: Microsecond

        Pulsepar1; % [phase, tipAngle] pair, for example: [0, pi/2] pulse
        Pulsepar2; % [phase, tipAngle] pair, for example: [0, pi] pulse
    PulseOperator1 = zeros(16,16); % Pulse superoperator in Liouville space, for example: pi/2
pulse
    PulseOperator2 = zeros(16,16); % Pulse superoperator in Liouville space, for example: pi
pulse

  end % properties

  methods
    function this = Pulse_class(hamiltonian, density, pulsetime0, pulsepar1, pulsepar2)
      this.Hamiltonian = hamiltonian;
      this.Density = density; % class
      this.Pulsetime0 = pulsetime0; % (~5 ns for pi/2 pulse, LPF p3667) unit: Microsecond
      this.Pulsepar1 = pulsepar1; % [phase, tipAngle] pair, for example: [0, pi/2] pulse
      this.Pulsepar2 = pulsepar2; % [phase, tipAngle] pair, for example: [0, pi] pulse
    end % Pulse_class constructor

    % pre calculations
    function Init(this)
      this.PulseOperator1 = this.CalPulseOperator(this.Pulsepar1);
      this.PulseOperator2 = this.CalPulseOperator(this.Pulsepar2);
    end

    % Calculate pulse superoperator in Liouville space
    function [pulseOperator] = CalPulseOperator(this, pulsepar)
      phase = pulsepar(1);
      tipAngle = pulsepar(2); % tipAngle = Pulsetime0 * B1* gyromagneticRatio, [LPF A7]

      tp = this.Pulsetime0; % pulse time
      omiga = tipAngle / tp; % angle velocity

      % In |SzIz> basis
      m = 4;
      h1 = zeros(m,m); % (= epsilon) irradiating microwave pulse, in Hilbert space and |SzIz>
basis

      % Following [LPF A7]
      h1(1,3) = 0.5* omiga * exp(-1i*phase);
      h1(2,4) = h1(1,3);

      h1(3,1) = 0.5* omiga * exp(1i*phase);
      h1(4,2) = h1(3,1);

      h1 = tp*h1;

      pulseH = this.Hamiltonian.SzIzToH0BasisH(h1);
      % pulseH in Liouville space
```

```
            pulseLiouville = kron(eye(4),pulseH) -  kron(ctranspose(pulseH), eye(4));

            % Pulse superoperator in L0 basis
            pulseOperator = expm(1i*pulseLiouville);

            %in Liouville normal mode basis
            pulseOperator = this.Hamiltonian.L0ToNormalBasis(pulseOperator);
        end % function CalPulseOperator

        % Calculate density evolve under a pulseOperator
        function DensityEvolve(this, pulsepar)
            %phase = pulsepar(1);
            tipAngle = pulsepar(2);

            switch tipAngle
                case pi/2.0 % pi/2 pulse
                    pulseOperator = round(this.PulseOperator1*10e12)/(10e12);
                case pi % pi pulse
                    pulseOperator = this.PulseOperator2;
                otherwise
                    error('No such pulse type!')
            end %switch path

            this.Density.DensityL = pulseOperator * this.Density.DensityL;

        end %function DensityEvolve

    end % methods

end %classdef Pulse_class
```

10. Relaxation_class

```
% LPF Eqs. A20 - A22
classdef Relaxation_class < handle
    properties
        Relaxation = zeros(16, 16); % Relaxation operator in Liouville space and |SzIz> basis

        T2e = 1.0; % Off-diagonal electron spin–spin relaxation, unit: microsecond
        T2n = 1.0; % Off-diagonal nuclear spin–spin relaxation, unit: microsecond

        We = -1.0; % Diagonal electron spin–spin relaxation, unit: microsecond
        Wn = -1.0; % Diagonal nuclear spin–spin relaxation, unit: microsecond

        Wx = -1.0; % Diagonal electron nuclear spin–spin relaxation, unit: microsecond
        Wy = -1.0; % Diagonal electron nuclear spin–spin relaxation, unit: microsecond
        Whe = -1.0; % Heisenberg exchange relaxation

        WithRelaxation;
        WithRelaxationMatrix = true;

    end %properties

    methods

        %Suppose, at initial state, the spin is in in z-axis
        function this = Relaxation_class(t2e, t2n, we, wn, wx, wy, whe,
withRelaxation,withRelaxationMatrix)
            relaxation = zeros(16, 16); % Relaxation operator in Liouville space and |SzIz> basis

            this.T2e = t2e; % T2 electron spin–spin relaxation, unit: microsecond
            this.T2n = t2n; % T2 nuclear spin–spin relaxation, unit: microsecond

            this.We = we; % Diagonal electron spin–spin relaxation, unit: microsecond
            this.Wn = wn;% Diagonal nuclear spin–spin relaxation, unit: microsecond

            this.Wx = wx;% Diagonal electron nuclear spin–spin relaxation, unit: microsecond
            this.Wy = wy; % Diagonal electron nuclear spin–spin relaxation, unit: microsecond

            this.Whe = whe; % Heisenberg exchange relaxation

            this.WithRelaxation = withRelaxation;
            this.WithRelaxation=withRelaxationMatrix;
            RT2e = -1.0 / t2e;
            RT2n = -1.0 / t2n;

            rt_ba = RT2n;
            rt_ca = RT2e;
            rt_da = RT2e;

            rt_ab = RT2n;
            rt_cb = RT2e;
            rt_db = RT2e;

            rt_ac = RT2e;
            rt_bc = RT2e;
            rt_dc = RT2n;

            rt_ad = RT2e;
            rt_bd = RT2e;
            rt_cd = RT2n;
```

```
w_ab = wn;
w_ba = wn;
w_cd = wn;
w_dc = wn;

w_ac = we;
w_ca = we;
w_bd = we;
w_db = we;

w_ad = wy;
w_da = wy;

w_bc = wx;
w_cb = wx;

w_ab = w_ab + whe;
w_ac = w_ac + whe;
w_ad = w_ad - whe;

w_ba = w_ba + whe;
w_bc = w_bc - whe;
w_bd = w_bd + whe;

w_ca = w_ca + whe;
w_cb = w_cb - whe;
w_cd = w_cd + whe;

w_da = w_da - whe;
w_db = w_db + whe;
w_dc = w_dc + whe;

w_aa = -(w_ab + w_ac + w_ad);
w_bb = -(w_ba + w_bc + w_bd);
w_cc = -(w_ca + w_cb + w_cd);
w_dd = -(w_da + w_db + w_dc);

relaxation(1,1) = w_aa;
relaxation(2,2) = rt_ba;
relaxation(3,3) = rt_ca;
relaxation(4,4) = rt_da;
relaxation(5,5) = rt_ab;
relaxation(6,6) = w_bb;
relaxation(7,7) = rt_cb;
relaxation(8,8) = rt_db;
```

```
            relaxation(9,9) = rt_ac;
            relaxation(10,10) = rt_bc;
            relaxation(11,11) = w_cc;
            relaxation(12,12) = rt_dc;
            relaxation(13,13) = rt_ad;
            relaxation(14,14) = rt_bd;
            relaxation(15,15) = rt_cd;
            relaxation(16,16) = w_dd;

            %relaxation(1,1) = w_aa;
            relaxation(1,6) = w_ab;
            relaxation(1,11) = w_ac;
            relaxation(1,16) = w_ad;

            relaxation(6,1) = w_ba;
            %relaxation(6,6) = w_bb;
            relaxation(6,11) = w_bc;
            relaxation(6,16) = w_bd;

            relaxation(11,1) = w_ca;
            relaxation(11,6) = w_cb;
            %relaxation(11,11) = w_cc;
            relaxation(11,16) = w_cd;

            relaxation(16,1) = w_da;
            relaxation(16,6) = w_db;
            relaxation(16,11) = w_dc;
            %relaxation(16,16) = w_dd;

            this.Relaxation = relaxation;
        end %  Relaxation_class constructor this
    end % methods

end %classdef Relaxation_class
```

11. Signal_class

```
% Calculate ESEEM signals according to a pathway
classdef Signal_class < handle
    properties

        Hamiltonian; % class
        Density; % class
        Pulse; % class
        Relaxation; % class
        FED; % class
        Pathway; % class
```

```
      SignalType; % % Sc+ or Sc-
      Tv1; % t1s
      Tv2; % t2s
      Euler_angle;
      SignalFile; % output

      Delta; %Gaussian inhomogeneous broadening effect, unit: MHz
      WithGaussian; % Gaussian inhomogeneous broadening effect

      S_cplus = zeros(4,4);  % Measurement matrix Sc+ = (Sx + iSy) in the original basis (|SzIz>)
      S_cminus = zeros(4,4); % Measurement matrix Sc- = (Sx - iSy) in the original basis (|SzIz>)

    end

    methods

      function this = Signal_class(hamiltonian, density, pulse, fED, relaxation, pathway, ...
          signalType, tv1, tv2, euler_angles, withGaussian, delta,signalFile)

        this.Hamiltonian = hamiltonian; % class
        this.Density = density; % class
        this.Pulse = pulse; % class
        this.FED = fED; % class
        this.Relaxation = relaxation; % class
        this.Pathway = pathway; % class

        this.SignalType = signalType; % Sc+ or Sc-

        this.Tv1 = tv1; % t1s
        this.Tv2 = tv2; % t2s
        this.Euler_angle = euler_angles;
        this.WithGaussian = withGaussian; % true or false
        this.Delta = delta; % for Gaussian broadening
        this.SignalFile = signalFile; % output

      end

      function Init(this, euler_angles)
        hamiltonian = this.Hamiltonian;
        hamiltonian.Init(euler_angles);

        this.Density.Init();
        this.Pulse.Init();

        % Sc- = (Sx + iSy) in the original basis (|SzIz>)
    this.S_cplus(1,3) = complex(1.0, 0.0);
    this.S_cplus(2,4) = complex(1.0, 0.0);
    %           this.S_cplus=this.Hamiltonian.SzIzToH0BasisH(this.S_cplus);
    %Sc+ = (Sx - iSy) in the original basis (|SzIz>)
    this.S_cminus(3,1) = complex(1.0, 0.0);
    this.S_cminus(4,2) = complex(1.0, 0.0);
    %           this.S_cminus=this.Hamiltonian.SzIzToH0BasisH(this.S_cminus);
end % Signal_class constructor

function Output2Pulses(this, pulses, spinPathway)

    % pulsepar(2): (phase, tipAngle) pairs
    tv1 = this.Tv1;
    tv2 = this.Tv2;

    n1 = length(this.Tv1);
    n2 = length(this.Tv2);
    signals = zeros(n1, n2);

    this.Init(this.Euler_angle);

    this.Density.SzIzToHbasis();
    this.Density.L0ToNormalBasis();

    this.Pulse.DensityEvolve(pulses{1});

    this.Density.NormalToL0Basis();

    this.Pathway.SelectPath(spinPathway(1)); % After pulse

    this.Density.L0ToNormalBasis();
    densityLst1 = this.Density.DensityL; % start point of dimension 1

    for i = 1: n1

        this.Density.DensityL = densityLst1;

        t1 = tv1(i);

        this.FED.DensityEvolve(t1);

        this.Pulse.DensityEvolve(pulses{2});
        this.Density.NormalToL0Basis();
        this.Pathway.SelectPath(spinPathway(2)); % After pulse

        this.Density.L0ToNormalBasis();
        this.FED.DensityEvolve(t1);

        densityLst2 = this.Density.DensityL; % start point of dimension 2

        for j = 1: n2

            this.Density.DensityL = densityLst2;

            t2 = tv2(j) ; %SECSY
            this.FED.DensityEvolve(t2);

            this.Density.NormalToL0Basis();

            signal = this.Measure ();
            signal = imag(signal); % real-dispersion, imag-absorption, abs-Complex Magnitude

            if(this.WithGaussian) % Gaussian inhomogeneous broadening effect
                delta = this.Delta;
                g = exp(-2 * (pi * delta * t2)^2); % Gaussian inhomogeneous broadening effect

                signal = g.*signal; % Gaussian inhomogeneous broadening effect
            end

            signals(i,j) = signal;

        end % for j = 1: n2

    end % for i = 1: n1

    dlmwrite(this.SignalFile, signals);

end %function Output2Pulses

function Output3Pulses(this, pulses, t, spinPathway)

    % pulsepar(2): (phase, tipAngle) pairs
    tv1 = this.Tv1;
    tv2 = this.Tv2;

    n1 = length(this.Tv1);
    n2 = length(this.Tv2);
    signals = zeros(n1, n2);

    this.Init(this.Euler_angle);

    this.Density.L0ToNormalBasis();
this.Pulse.DensityEvolve(pulses{1});
this.Density.NormalToL0Basis();

this.Pathway.SelectPath(spinPathway(1)); % After pulse
this.Density.L0ToNormalBasis();
densityLst1 = this.Density.DensityL; % start point of dimension 1

for i = 1: n1 % i = 1: n1

   this.Density.DensityL = densityLst1;

   t1 = tv1(i);
   this.FED.DensityEvolve(t1);

   %spin = 2;
   this.Pulse.DensityEvolve(pulses{2});
   this.Density.NormalToL0Basis();
   this.Pathway.SelectPath(spinPathway(2)); % After pulse
   this.Density.L0ToNormalBasis();
   this.FED.DensityEvolve(t);

   %spin = 1;
   this.Pulse.DensityEvolve(pulses{3});
   this.Density.NormalToL0Basis();
   this.Pathway.SelectPath(spinPathway(3)); % After pulse
   this.Density.L0ToNormalBasis();
   this.FED.DensityEvolve(t1);
   densityLst2 = this.Density.DensityL; % start point of dimension 2

   for j = 1: n2 %j = 1: n2

      this.Density.DensityL = densityLst2;

      t2 = tv2(j); %SECSY

      this.FED.DensityEvolve(t2);
      this.Density.NormalToL0Basis();

      signal = this.Measure ();
      signal = imag(signal); % real-dispersion, imag-absorption, abs-Complex Magnitude


      if(this.WithGaussian) % Gaussian inhomogeneous broadening effect
         delta = this.Delta;
         g = exp(-2 * (pi * delta * (t2))^2); % Gaussian inhomogeneous broadening effect
         signal = g*signal; % Gaussian inhomogeneous broadening effect
```

```
               end

               signals(i,j) = signal;

            end % for j = 1: n2

         end % for i = 1: n1

         dlmwrite(this.SignalFile, signals);
      end %function Output3Pulses

      function [signal] = Measure (this)
         this.Density.HToSzIzBasis();

         density = this.Density.GetDensityMat();

         switch this.SignalType
            case 1 %Sc-
               measurementMat = this.S_cplus;
            case 2 %Sc+
               measurementMat = this.S_cminus;
            otherwise
               error('No such Signal Type, thus output default Sc-');
         end %switch SignalType

         temp = measurementMat * (density);
         signal = trace(temp); % the sum of the diagonal elements of the matrix
      end % function Measure

   end % methods

end %classdef Signal_class < handle
```